

AÑO II - NUM. 45

MICROHOBBY SEMANAL

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES SINCLAIR
AÑO II - N.º 45

125 PTS.
Canarias 135 ptas.

EDITA HOBBY PRESS S.A

LONDRES
LAS NOVEDADES DE LA 8ª "PERSONAL COMPUTER SHOW"

PROGRAMAS
- EL BUZO
- BASKET
- LOS ANILLOS

NUEVO
"NIGHT SHADE", ULTIMATE RIZA EL RIZO

TRUCOS
CONVERSOR HEXADECIMAL EN CODIGO MAQUINA



Director Editorial
José I. Gómez-Centurión
Director Ejecutivo
Domingo Gómez
Subdirector
Gabriel Nieto
Redactor Jefe
Africa Pérez Tolosa
Diseño
Rosa María Capitel
Redacción
José María Díaz, Miguel Angel Hijosa, Fco. Javier Martín
Secretaria Redacción
Carmen Santamaría
Colaboradores
Jesús Alonso, Primitivo de Francisco, Rafael Prades, Miguel Sepúlveda, Sergio Martínez y J. M. Lazo
Fotografía
Javier Martínez, Carlos Candel
Portada
José María Ponce
Dibujos
J.R. Ballesteros, A. Perera, F.L. Frontán, Pejo, J.M. López Moreno, J. Igual, J.A. Calvo, Lóriga, Luisma, J. Olivares.

Edita
HOBBY PRESS, S.A.

Presidente
María Andrino
Consejero Delegado
José I. Gómez-Centurión

Jefe de Publicidad
Marisa Esteban
Publicidad Barcelona
José Galán Cortés, Tels.: 303 10 22 - 313 71 76.

Secretaria de Dirección
Marisa Cogorro

Suscripciones
M.ª Rosa González
M.ª del Mar Calzada

Redacción, Administración y Publicidad
La Granja, n.º 8
Polígono Industrial de Alcobendas
Tel.: 654 32 11
Telex: 49480 HOPR

Dto. Circulación
Carlos Peropadre

Distribución
Coedis, S.A. Valencia, 245
Barcelona

Imprime
Rotedic, S.A.
Carretera de Irún, Km. 12,450
Tel.: 734 15 00
Fotocomposición
Espacio y Punto, S.A.
Paseo de la Castellana, 268
Fotomecánica
Grof
Ezequiel Solana, 16
Depósito Legal:
M-36.598-1984

Representante para Argentina, Chile, Uruguay y Paraguay, Cía. Americana de Ediciones, S.R.L. Sud América, 1.532. Tel.: 21 24 64. 1209 BUENOS AIRES (Argentina).

Solicitado control OJD

MICROHOBBY ESTA SEMANA

AÑO II. N.º 45. 24 al 30 de septiembre de 1985
125 ptas. (Sobretasa Canarias 10 ptas.)





¡¡ESTUVIMOS en la «Personal Computer Show» de Londres!!

PREMIADOS HOBBY-SUERTE

OSCAR GOMEZ RAMOS. Navas de Tolosa, 2, Bj Dr. Valle de Trápaga (VIZCAYA).
Cinta de programas (5.ª Cat.)

RAMON MARQUEZ MONTERO. Alameda S. Antón, 27, Bl Iz, 6.º Cartagena (MURCIA).
Suscripción a Microhobby Semanal por un año (4.ª Cat.)

FRANCISCO ROBLES. Pedro Maseveu, 7, ESC. 1, 1.º D. Oviedo (ASTURIAS)
Cinta de programas (5.ª Cat.)

PEDRO BRAVO. S. Martín de Porres, 20 (MADRID).
Cinta de programas (5.ª Cat.)

JOSE LUIS MICHE GONZALEZ. Ronda de Pío XII, 18 (SEVILLA).
Cinta de programas (5.ª Cat.)

JOSE LUIS RODRIGUEZ SANCHEZ. Ctra. de Montiel, 22. Villahermosa (CIUDAD REAL).
Cinta de programas (5.ª Cat.)

MANUEL SALDAÑA PEREZ. Palomar, 29. Puerto-Llano (CIUDAD REAL).
Cinta de programas (5.ª Cat.)

ANGEL ANTON RODRIGUEZ. S. Clemente, 10, 1.º A. Santiago de Compostela (LA CORUÑA).
Cinta de programas (5.ª Cat.)

MANUEL DIEZ ALVAREZ. Cea Bermúdez, 42, 3. Gijón (ASTURIAS).
Suscripción a Microhobby Semanal por un año (4.ª Cat.)

DALMIRO GONZALEZ. Martínez Maldonada, 71. (MALAGA).
Cinta de programas (5.ª Cat.)

ABELLA CRUZ. Chile, 2, 4, 5.º Izda. Ferrol (LA CORUÑA).
Suscripción a Microhobby Semanal por un año (4.ª Cat.)

JESUS LOPEZ MOURIZ. Autonomía, 83, 1.º, 1.º Tarrasa (BARCELONA).
Cinta de programas (5.ª Cat.)

VICENTE VELA NIETO. Puerta de Alancos, 3, 6.º. (CIUDAD REAL).
Cinta de programas (5.ª Cat.)

ENRIQUE GARCIA SANCHEZ. Avda. del Sur, 5, 1.º Carlets (VALENCIA).
Cinta de programas (5.ª Cat.)

FRANCISCO YUNA BELTRAN. Salvador Rubio, 1. (VALENCIA).
Cinta de programas (5.ª Cat.)

FRANCISCO HERNANDEZ YAÑEZ. Avefría, 16, 3.º D. (MADRID).
Cinta de programas (5.ª Cat.)



Durante los días 4 al 8 de septiembre se ha celebrado en el Olympia de Londres, la 8 PERSONAL COMPUTER SHOW, considerada como la Feria de Ordenadores Personales más importante de Europa, y que abarca todo un extenso mercado de aplicaciones, educación y juegos. Hasta allí nos desplazamos para mantener puntualmente informados a nuestros lectores de todas las novedades producidas en este campo, y más concretamente, las relacionadas con ordenadores Sinclair.

Este año, octavo de su celebración, ha reunido a más de doscientas compañías especializadas en las diferentes áreas que cubre la muestra.


Aspecto general de la feria.

LA AVENTURA A TU ALCANCE

Siguiendo la idea de lo que parece ser se ha convertido en el mayor negocio de software, la adaptación de películas al ordenador, no podía faltar lo último de Steven Speilberg, «The Goonies». Por contar con uno de los más apreciados condimentos a la hora de programar, la aventura, la obra de este joven director ya fue un éxito en este mundo informático con «Gremlins» y, ahora, estamos seguros que no será menor con esta nueva adaptación.

U.S. Gold es la compañía embarcada en este negocio seguro que pronto podremos disfrutar en España.



EL QL EVOLUCIONA: *Más potencia a menor precio.*

Sinclair refuerza notablemente el atractivo de su QL en base a una drástica reducción del precio y a la incorporación de nuevos periféricos y paquetes de aplicaciones, desarrollados por Eidersoft y Psion.

La mayoría de los visitantes acudían al stand atraídos por la sorprendente rebaja sufrida por el QL, que ha pasado a valer 199 libras, en lugar de las 399 que empezó costando, lo que supone un 50% de ahorro. Esta rebaja se interpreta en medios especializados como un intento de competir con otras máquinas de similares prestaciones, a la vez que se pretende aumentar el número de QLs en el mercado, lo que aseguraría su supervivencia. Esto es así porque los fabricantes de Software no están dispuestos a desarrollar programas para ordenadores de poca difusión.



El stand de Sinclair, uno de los más visitados.

Como gran novedad, presentaron las unidades de Disco para el QL de 3,5" fabricadas por la compañía de hardware Micro Peripherals Ltd. (aunque esta vez supervisadas directamente por Sinclair).

Dichas unidades están capacitadas para almacenar 720 K por disco (formateado a doble cara) y son controladas por el QL Double Density Disk Interface, también de la misma casa. El promedio de tiempo de acceso es de, aproximadamente, un segundo.

El sistema completo será distribuido en exclusiva para España por Investrónica.

TARJETA DE SOFT

Sofcard es un aparato de memoria sólida que sorprende por su diseño compacto y por la gran variedad de usos.

Las nuevas tecnologías han permitido crear un IC chip de reducidas dimensiones y superficie plana que, además, no pierde capacidad de memoria.

Impreso en una tarjeta de tamaño de las de crédito, tenemos 256 Kb de capacidad de memoria y un megabyte en la versión que estará disponible próximamente. Sofcard puede ser también usado como sustitutivo de un cassette o un disco, y cuando se necesite una expansión de memoria.

Sus aplicaciones son múltiples, pero el terreno donde se muestra más revolucionario es en el de los juegos, ya que permitirá disponer de un soporte fiable, barato, de velocidad de carga casi instantánea y que además, ocupa un reducido espacio.

Otra de sus habilidades es la de permitir la creación de imágenes en color y la información puede ser impresa por ambas caras.

Es adaptable a diferentes tipos de ordenadores a través de las zonas que utilizan éstos para los cartuchos de ROM.

Se trata, en definitiva, de una tarjeta revolucionaria tanto por su precio como por sus aplicaciones, que supondrá un claro avance tecnológico y sobre todo, un estupendo soporte para todo tipo de aplicaciones.



OTRA COMPAÑIA DE SOFTWARE

Una nueva compañía hace su aparición en la escena del Software. Estamos hablando de Electric Dreams. Y lo ha hecho con dos juegos que, a juicio de muchos, son bastante revolucionarios en muchos aspectos. Sobre todo «I, OF THE MASK», que utiliza una nueva generación de las técnicas gráficas en 3 dimensiones, con algunas secuencias de una especie de cabeza geométrica que gira alrededor de sí mismo suspendida en el espacio. El personaje del juego se desenvuelve en una estructura laberíntica tridimensional de la que será muy difícil salir.

El autor es Sany White, el legendario creador del Ant Atack (Hormigas), que además es el primer programador que investigó los sistemas tridimensionales en los juegos para Spectrum, por cierto con bastante éxito.

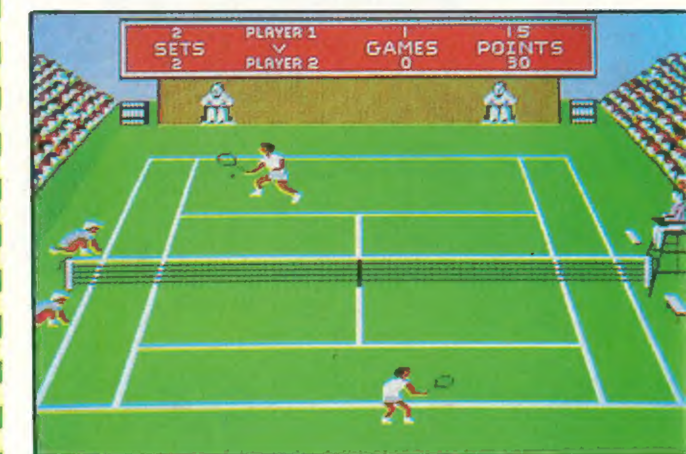
A diferencia de otros programas, las pantallas por las que vamos pasando no se encuentran archivadas en un banco de datos, sino que se forman esporádicamente partiendo de logaritmos matemáticos.

Otra de las innovaciones que incorpora, es la utilización de sombreados en la formación de gráficos con el fin de evitar la molesta mezcla de atributos tan peculiar del Spectrum.

El otro programa se llama RIDDLER'S DEN, una aventura basada en un simpático personaje que es una especie de hombre elefante.

MATCH POINT, PARA EL QL

Por fin, y aunque parezca mentira, alguien ha pensado que sería muy interesante plantearse que el QL puede ser también un magnífico ordenador de juegos. Ese alguien se llama PSION y el programa no es otro que el «archiconocido» Match Point, pero en esta ocasión en su versión para este ordenador.



Se ha aprovechado bastante bien la potencia gráfica del QL para dotar al programa de una vistosidad y unos efectos que superan, incluso, a los de la primera versión.

El programa ya está disponible en el Reino Unido y fue presentado también en la feria de Londres.

MICROPANORAMA

TWINRITER 5



Twinriter 5 es la nueva impresora de Brother, que ha sido presentada recientemente en la Personal Computer Show.

El método de impresión que usa es el de matriz de puntos y tiene una velocidad máxima de 40 caracteres por segundo, con un sistema de carga bidireccional.

Otros datos de interés son: la capacidad de copia, la posibilidad de incorporar un Interface RS 232 además del Centronics, es compatible con IBM y el buffer de 3 Kb.

EL SEÑOR DE LOS ANILLOS

Héroes y villanos, aventuras y excitación, colorido y animación son los mejores argumentos para definir una de las películas más taquilleras de los últimos años, «El señor de los Anillos», que, como ya viene siendo habitual, va a deleitar a muchos en la nueva versión creada para ordenador.

La compañía encargada de llevar a cabo este evento, de lo más plausible, ha sido MELBOURNE HOUSE basándose en el trabajo de Philip Mitchell que ya



nos deleitara anteriormente con «The Hobbit» y estará disponible para Spectrum 48 K, Commodore 64 y Amstrad, según afirmarón en la feria.

Esperemos que sea pronto.

CONFIRMADO:

EL SUPERSPECTRUM DE 128 K SE PRESENTA EN SONIMAG

Los insistentes rumores que circulaban en los últimos días sobre la aparición de una nueva máquina Sinclair, acaban de confirmarse. El Spectrum 128 K es ya un hecho.

La presentación en SONIMAG del nuevo ordenador se hace como auténtica primicia mundial, ya que por el momento sólo saldrá en España. La venta en Inglaterra se ha pospuesto para el próximo año por exigencias de mercado ya que el amplio stock de Spectrum y Spectrum + existente en el Reino Unido desaconsejan la operación.

El tamaño es similar al del Spectrum+ y quizás, lo más significativo del nuevo modelo sea que incorpora un teclado numérico auxiliar que se acopla mediante un conector al principal.

La exclusiva de España se debe a la circunstancia de haber sido desarrollado conjuntamente por SINCLAIR e INVESTRONICA, su distribuidor oficial en nuestro país.

Como consecuencia de esto, el nuevo Spectrum lleva incorporada la letra «ñ», cumpliendo además todos los requisitos para adecuarse a la nueva normativa del Decreto Ley sobre homologación de ordenadores.

El teclado incluye inscripciones traducidas (en la medida de lo posible) como pueden ser los colores: blanco, azul, etc. en lugar de white, blue, etc.

Otro detalle importante es que no va a ser importado sino producido y fabricado íntegramente en España por INVESTRONICA.

La compatibilidad será total con el software existente y los primeros paquetes desarrollados específicamente serán íntegramente en castellano.

TRUCOS

¡LISTADO FUERA!

Para evitar que salga el listado en pantalla, algo que puede no ser conveniente ante miradas curiosas, German Bravo, nos ha enviado un POKE con el que se conseguirá borrarlo sin que se bloquee.

POKE 23659,255

Una advertencia, se debe incluir en la primera línea una vez que el programa se haya terminado y verificado.

DE DOS EN DOS

Ixidor Zarrañaga ha querido colaborar en esta sección, no con uno, sino con dos trucos que, según comenta en su carta, espera que nos sean de utilidad. Eso lo veréis vosotros.

El primero de ellos es para hacer BREAK en un programa tecleando RANDOMIZE USR 1357.

Por ejemplo: 1Ø INPUT a: IF a= 12345 THEN RANDOMIZE USR 1357.

Si en medio de un programa hecho por nosotros ponemos el ejemplo anterior y el número 12345 sólo lo conocemos nosotros y el programa lo hemos protegido, lo podemos cortar.

El segundo truco sirve para hacer NEW sin que el PAPER se oscurezca.

Tecleando RANDOMIZE USR 4657 equivale a la sentencia NEW.

Si en algún caso quisiéramos poner la frase «Sinclair Research Ltd.» se puede poner RANDOMIZE USR 4756, saliendo en la parte superior de la pantalla.

Si, por el contrario, queremos que salga en cualquier lado de la pantalla, habrá que poner, por ejemplo:

PRINT AT 1Ø,Ø;USR 4756

Y para que aparezca en su lugar correcto:

PRINT # Ø; USR 4756

Esto último no borra un hipotético programa que estuviese en memoria.



CARGADOR HEXADECIMAL EN CODIGO MAQUINA

Una vez más el lenguaje máquina viene en nuestro auxilio para implementar una pequeña rutina de gran utilidad para los programadores, y que puede emplearse desde Basic sin ningún problema.

Se trata de representar en hexadecimal cualquier número decimal, de Ø a 65535.

Para ello, como puede observarse en la línea número 2Ø del programa cargador Basic, introducimos nuestro número decimal en la variable SEED, mediante la instrucción RANDOMIZE.

Luego, basta llamar a la rutina en máquina que hace el trabajo duro e instantáneamente aparecerá el número en la pantalla.

Como casi siempre, proponemos como dirección de ensamblado el buffer de impresora, utilizado en el programa cargador.

Para los que posean un ensamblador o estén interesados en averiguar cómo funciona la rutina, proporcionamos también la rutina escrita en lenguaje ensamblador.

```
  10 INPUT "NUMERO ";CIF
  20 RANDOMIZE CIF: RANDOMIZE US
R 23296
  30 GO TO 10
 100 FOR N=23296 TO 23346: READ
A: POKE N,A: NEXT N
  110 DATA 253,203,2,134,237,91,
118,92,74,175,186,40,3,205,24,91
,75,205,24,91,62,13,215,201,121,
230,240,203,63,203,63,203,63,203
,63,205,41,91,121,230,15,198,48,
254,58,56,2,198,7,215,201
```

```
 10          ORG   23296
 20 ;
 30          RES   0,(IY+2)
 40          LD    DE,(23670)
 50          LD    C,D
 60          XOR   A
 70          CP    D
 80          JR    Z,BAJO
 90          CALL  HEXA
100 BAJO     LD    C,E
110          CALL  HEXA
120          LD    A,13
130          RST   16
140          RET
150 HEXA     LD    A,C
160          AND   #F0
170          SRL   A
180          SRL   A
190          SRL   A
200          SRL   A
210          CALL  LOW
220          LD    A,C
230          AND   #F
240 LOW      ADD   A,"0"
250          CP    58
260          JR    C,PR
270          ADD   A,7
280 PR       RST   16
290          RET
```

En este espacio también tienen cabida los trucos que nuestros lectores quieran proponer. Para ello, no tienen más que enviarlos por correo a MICROHOBBY, C/ La Granja, 8. Polígono Industrial de Alcobendas (Madrid).

LOS ANILLOS

Carlos CERVERA

Spectrum 48 K

La imaginación al alcance de tu mano, o mejor de tu dedo, podría ser la frase ideal para presentar este original juego cargado de fantasía.

Y esta fantasía va a hacerte falta si te sumerges en él, ya que estaremos representados por un anillo al que hemos de conducir a través de siniestros túneles y cuevas en busca de un dedo, el de su amo.

La tarea no es fácil, desde luego, si tenemos en cuenta que varios dedos más estarán esparcidos por este fantasmagórico sitio, provocándonos alguna que otra confusión a la hora de reconocer el buscado. Por otro lado, al profanar las cuevas, las uñas guardianas, enfurecidas, nos perseguirán y unas bolas saltarinas intentarán aplastarnos a su paso.

Muchas sorpresas encierra este divertido juego. Prueba a superarlas.

NOTAS GRAFICAS

A B C D E F G H I J K L M N

```
   6 PAPER 0: BORDER 0: BRIGHT 1
: INK 7: CLS
   7 RESTORE 9900: GO SUB 9900:
LET rec=0
   8 PRINT AT 10,10;"ANILLO LOCO
": PLOT 0,0: DRAW 255,0: DRAW 0,
175: DRAW -255,0: DRAW 0,-175: P
RINT AT 1,1;"NECESITAS INSTRUCCI
ONES? (s/n)": FOR f=0 TO 40: BEE
P .008,f: NEXT f
   9 IF INKEY$="s" THEN GO TO 12
  10 IF INKEY$="n" THEN GO TO 16
  11 GO TO 9
  12 POKE USR "n"+0,BIN 00111100
: INK 0: CLS : PRINT AT 0,8;"INS
TRUCCIONES";AT 1,8;"-----------
-";AT 4,0;"TU ERES UN ANILLO INF
ILTRADO EN LAS GALERIAS DE UNA C
UEVA.ESTAS EN ESTA CUEVA PUES HA
S PERDIDO  EL DEDO DE TU AMO.AL
VER TANTOS DEDOS ALLI DECIDES RE
CUPERARLOS PERO TU MISION SE VER
A OBSTRUIDAPOR LA PRESENCIA DE L
AS CRUELES U~AS DE LAS CUEVAS

          LAS TECLAS SON:    Q..
.ARRIBA                      A..
.ABAJO                       O..
.IZQUIERDA                   P..
.DERECHA
          PULSA UNA TECLA PARA
CONTINUAR.."
  13 INK 7: FOR f=0 TO 21: FOR n
=0 TO 31: IF CODE SCREEN$ (f,n)=
0 THEN PRINT AT f,n;"⊟": BEEP .0
1,10: NEXT n: NEXT f
  14 PRINT AT f,n;SCREEN$ (f,n):
 BEEP .01,10: NEXT n: NEXT f
  15 PRINT AT 21,15;"O": PAUSE 1
0: PRINT AT 21,15;"0": PAUSE 10:
 PRINT AT 21,15;"|": PAUSE 10: P
RINT AT 21,15;"0": PAUSE 10: IF
INKEY$="" THEN GO TO 15
  16 PRINT AT 21,0;"
                    ": RESTORE 1
7: FOR f=0 TO 20: POKE 23692,255
: READ a,a$: PRINT INK a;a$: NEX
T f
  17 DATA 6,"[redacted]",6,"[redacted]",6,"[redacted]",6,"[redacted]"
  18 DATA 6,"[redacted]",6,"[redacted]"
  19 DATA 6,"[redacted]"
  20 DATA 6,"[redacted]",6,"[redacted]"
  21 DATA 6,"[redacted]"
  22 DATA 6,"[redacted]",6,"[redacted]"
  23 DATA 6,"[redacted]",6,"[redacted]",0,"
                          ",5,"
   Por Carlos Cervera
  24 DATA 0,"
              ",0,"
                    ",4,"Pulsa una
 tecla para empezar...."
  25 DATA 0,"
              ",0,"
                    "
  26 IF INKEY$="" THEN GO TO 26
  30 LET vid=3: LET punt=0
 100 REM *** NIVEL 1 ***
 110 LET nivel=100: LET pas=0: L
ET x=1: LET y=20: LET x1=10: LET
 y1=5: LET n=1
 120 CLS : PRINT INK 5;AT 0,0;"P
UNTOS:";punt; INK 6;AT 0,12;"VID
AS:";vid; INK 4;AT 0,21;"RECORD:
";rec
 130 PRINT INK 6;AT 4,0;"●";AT 5
,0;"■"
 140 PRINT INK 2;AT 6,0;"▼▼▼▼▼▼▼▼
▼▼▼▼▼▼"; INK 7;"⊟"; INK 2;"▼▼▼▼▼▼
▼▼▼▼▼▼▼▼▼▼"; INK 7;"⊟";AT 7,13
;"⊟";AT 8,13;"⊟"
 150 PRINT INK 7;AT 7,31;"⊟";AT
8,31;"⊟"
 160 PRINT INK 2;AT 9,0;"▼▼▼"; I
NK 7;"⊟"; INK 2;"▼▼▼▼▼▼▼▼▼▼▼▼▼▼";
 INK 7;"⊟"; INK 2;"▼▼▼▼▼▼▼▼▼▼▼▼▼▼
▼"
 170 PRINT INK 7;AT 10,3;"⊟";AT
10,17;"⊟";AT 11,3;"⊟";AT 11,17;"
⊟"
 180 PRINT INK 2;AT 12,0;"▼▼▼▼▼▼
▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼"; INK 7;"⊟"; IN
K 2;"▼▼▼▼▼▼▼▼▼"
 190 PRINT INK 7;AT 13,22;"⊟";AT
14,22;"⊟"
 200 PRINT INK 7;AT 15,0;"⊟"; IN
K 2;"▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼
▼▼▼▼"; INK 7;AT 16,0;"⊟";AT 17,
0;"⊟"; INK 2;AT 18,0;"▼▼▼▼▼▼▼▼▼
▼▼▼▼▼▼▼▼▼▼▼"; INK 7;"⊟"; INK 2;"▼
▼▼▼▼▼▼▼▼▼▼▼▼"
 210 PRINT INK 7;AT 19,20;"⊟";AT
20,20;"⊟"
 220 PRINT INK 2;AT 21,0;"▼▼▼▼▼▼
▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼"
 300 LET a$="O0|0"
 310 IF INKEY$="o" THEN IF x>0 T
HEN IF y=20 OR y=17 OR y=14 OR y
=11 OR y=8 OR y=5 THEN LET x=x-1
: PRINT AT y,x;a$(n);AT y,x+1;"
"
 311 IF INKEY$="q" THEN IF CODE
SCREEN$ (y-1,x)=0 OR y=18 OR y=1
5 OR y=12 OR y=9 OR y=6 THEN LET
 y=y-1: PRINT AT y,x;a$(n);AT y+
1,x;"⊟"
 312 IF INKEY$="a" THEN IF CODE
SCREEN$ (y-1,x)=32 THEN IF CODE
SCREEN$ (y+5,x)=32 THEN IF CODE
SCREEN$ (y+4,x)=0 THEN IF CODE S
CREEN$ (y+3,x)=0 THEN LET y=y+1:
 PRINT AT y,x;a$(n);AT y-1,x;" "
 313 IF INKEY$="a" THEN IF CODE
SCREEN$ (y-1,x)=32 THEN IF CODE
SCREEN$ (y+4,x)=32 THEN LET y=y+
1: PRINT AT y,x;a$(n);AT y-1,x;"
⊟"
 314 IF INKEY$="a" THEN IF CODE
SCREEN$ (y+1,x)=0 THEN IF CODE S
CREEN$ (y+2,x)=0 THEN LET y=y+1:
 PRINT AT y,x;a$(n);AT y-1,x;"⊟"
 320 IF INKEY$="p" THEN IF x<31
THEN IF y=20 OR y=17 OR y=14 OR
y=11 OR y=8 OR y=5 THEN LET x=x+
1: PRINT AT y,x;a$(n);AT y,x-1;"
"
 321 IF a$(n)="O" OR a$(n)="|" T
HEN GO TO 350
 325 IF x>x1 AND x1<31 THEN LET
x1=x1+1: PRINT INK 5;AT y1,x1;"●
";AT y1,x1-1;" "
 326 IF x<x1 AND x1>1 THEN LET x
1=x1-1: PRINT INK 5;AT y1,x1;"●"
;AT y1,x1+1;" "
 327 IF y1<y THEN IF CODE SCREEN
$ (y1+1,x1)=0 THEN IF CODE SCREE
N$ (y1+2,x1)=0 THEN LET pas=1: L
ET y1=y1+1: PRINT AT y1,x1; INK
5;"●"; INK 7;AT y1-1,x1;" ": IF
x1=x AND y1=y THEN GO TO 8900
 328 IF pas=1 THEN LET y1=y1+1:
PRINT AT y1,x1; INK 5;"●";AT y1-
1,x1; INK 7;"⊟": IF x1=x AND y1=
y THEN GO TO 8900
 329 IF pas=1 THEN LET y1=y1+1:
PRINT AT y1,x1; INK 5;"●"; INK 7
;AT y1-1,x1;"⊟": LET pas=0
 350 IF CODE SCREEN$ (20,20)<>0
THEN PRINT AT 20,20;"⊟"
 360 IF CODE SCREEN$ (17,0)<>0 T
HEN PRINT AT 17,0;"⊟"
 361 IF CODE SCREEN$ (8,13)<>0 T
HEN PRINT AT 8,13;"⊟"
 365 IF CODE SCREEN$ (14,22)<>0
THEN PRINT AT 14,22;"⊟"
 370 IF CODE SCREEN$ (11,3)<>0 T
HEN PRINT AT 11,3;"⊟"
 375 IF CODE SCREEN$ (11,17)<>0
THEN PRINT AT 11,17;"⊟"
 380 IF CODE SCREEN$ (8,31)<>0 T
HEN PRINT AT 8,31;"⊟"
 390 IF ATTR (y,x-1)=70 THEN LET
 punt=punt+100: LET pant=1: LET
nivel=500: GO SUB 8000
 395 IF x=x1 AND y=y1 THEN GO TO
 8900
 400 PRINT AT y,x;a$(n): LET n=n
+1: IF n=5 THEN LET n=1
 401 GO TO 310
 500 REM *** NIVEL 2 ***
 501 LET a$="O0|0": LET n=1
 510 CLS : PRINT INK 5;AT 0,0;"P
UNTOS:";punt; INK 6;AT 0,12;"VID
AS:";vid; INK 4;AT 0,21;"RECORD:
";rec
 511 INK 5
 520 PRINT AT 5,0;"[redacted]";AT 6,0;"[redacted]"
 521 PRINT AT 7,0;"[redacted]"
 522 PRINT "[redacted]"
 523 PRINT "[redacted]"
 524 PRINT "[redacted]"
 525 PRINT "[redacted]"
 526 PRINT "[redacted]"
 527 PRINT "[redacted]"
 528 INK 7
 530 PRINT AT 20,31; INK 6;"●"
 531 PRINT INK 3;AT 7,29;"●";AT
8,29;"■";AT 9,21;"●";AT 10,21;"■
"
 532 LET x=1: LET y=20: LET x1=1
6: LET y1=6: LET x2=18: LET y2=1
6
 540 IF INKEY$="q" THEN IF CODE
SCREEN$ (y-1,x)<>0 THEN LET y=y-
1: PRINT AT y,x;a$(n);AT y+1,x;"
"
 541 IF INKEY$="a" THEN IF CODE
SCREEN$ (y+1,x)=32 THEN LET y=y+
1: PRINT AT y,x;a$(n);AT y-1,x;"
"
 542 IF INKEY$="p" THEN IF CODE
SCREEN$ (y,x+1)=32 THEN LET x=x+
1: PRINT AT y,x;a$(n);AT y,x-1;"
"
 543 IF INKEY$="o" THEN IF CODE
SCREEN$ (y,x-1)=32 THEN LET x=x-
1: PRINT AT y,x;a$(n);AT y,x+1;"
"
 544 IF INT (RND*30)=0 THEN GO T
O 558
 548 IF x>x1 THEN IF CODE SCREEN
$ (y1,x1+1)=32 THEN LET x1=x1+1:
 PRINT INK 4;AT y1,x1;"●"; INK 0
;AT y1,x1-1;" "
 549 IF x<x1 THEN IF CODE SCREEN
$ (y1,x1-1)=32 THEN LET x1=x1-1:
 PRINT INK 4;AT y1,x1;"●"; INK 0
;AT y1,x1+1;" ":
 550 IF y<y1 THEN IF CODE SCREEN
$ (y1-1,x1)=32 THEN LET y1=y1-1:
 PRINT INK 4;AT y1,x1;"●"; INK 0
;AT y1+1,x1;" "
 551 IF y>y1 THEN IF CODE SCREEN
$ (y1+1,x1)=32 THEN LET y1=y1+1:
 PRINT INK 4;AT y1,x1;"●"; INK 0
;AT y1-1,x1;" "
 552 IF x>x2 THEN IF CODE SCREEN
$ (y2,x2+1)=32 THEN LET x2=x2+1:
 PRINT INK 4;AT y2,x2;"●"; INK 0
;AT y2,x2-1;" "
 553 IF x<x2 THEN IF CODE SCREEN
$ (y2,x2-1)=32 THEN LET x2=x2-1:
 PRINT INK 4;AT y2,x2;"●"; INK 0
;AT y2,x2+1;" "
 554 IF y<y2 THEN IF CODE SCREEN
$ (y2-1,x2)=32 THEN LET y2=y2-1:
 PRINT INK 4;AT y2,x2;"●"; INK 0
;AT y2+1,x2;" "
 555 IF y>y2 THEN IF CODE SCREEN
$ (y2+1,x2)=32 THEN LET y2=y2+1:
 PRINT INK 4;AT y2,x2;"●"; INK 0
;AT y2-1,x2;" "
 558 IF ATTR (y+1,x)=68 OR ATTR
(y-1,x)=68 OR ATTR (y,x-1)=68 OR
 ATTR (y,x+1)=68 THEN GO SUB 890
0
 560 IF ATTR (y+1,x)=67 OR ATTR
(y,x+1)=67 OR ATTR (y,x-1)=67 TH
EN IF x>23 THEN LET punt=punt+10
0: PRINT INK 0;AT 7,29;" ";AT 8,
29;" ": PRINT INK 5;AT 0,7;punt:
 IF CODE SCREEN$ (9,21)=32 THEN
PRINT AT 20,31;" "
 561 IF ATTR (y+1,x)=67 OR ATTR
(y,x+1)=67 OR ATTR (y,x-1)=67 TH
EN IF x<23 THEN LET punt=punt+10
0: PRINT INK 0;AT 9,21;" ";AT 10
,21;" ": PRINT INK 5;AT 0,7;punt
: IF CODE SCREEN$ (7,29)=32 THEN
 PRINT AT 20,31;" "
 570 IF x=31 THEN LET pant=2: LE
T nivel=600: GO SUB 8000
 580 PRINT AT y,x;a$(n): LET n=n
+1: IF n=5 THEN LET n=1
 581 GO TO 540
 600 REM ** NIVEL 3 **
 601 CLS : PRINT INK 5;AT 0,0;"P
UNTOS:";punt; INK 6;AT 0,12;"VID
AS:";vid; INK 4;AT 0,21;"RECORD:
";rec
 602 INK 3
 610 LET n=21: LET c=4: LET b=1:
 LET a$="[redacted]"
 620 PRINT AT n,c;a$(b TO ): LET
 b=b+4: LET c=c+4: LET n=n-1: IF
 b>28 THEN GO TO 635
 630 GO TO 620
 635 LET n=21: LET b=0
 640 PRINT AT n,b;"▲▲▲▲": LET b=
b+4: LET n=n-1: IF b>28 THEN GO
TO 645
 641 GO TO 640
 645 LET n=4: LET b=28
 650 PRINT AT n,0;a$(2 TO b);"■"
: LET b=b-4: LET n=n+1: IF b=0 T
HEN GO TO 655
 651 GO TO 650
 655 LET n=4: LET b=31-3
 660 PRINT AT n,b;"▼▼▼▼": LET b=
b-4: LET n=n+1: IF b<0 THEN GO T
O 670
 665 GO TO 660
 670 LET x=0: LET y=15: LET a$="
O0|0": LET n=1
 680 LET x1=10: LET y1=14: LET x
2=24: LET y2=10
 681 FOR f=5 TO 13: PRINT AT f,3
1; INK 1;"■": NEXT f
 682 INK 5: PRINT AT 13,27;"●";A
T 14,27;"■";AT 15,19;"●";AT 16,1
9;"■"
 689 INK 7
 690 IF INKEY$="q" THEN IF CODE
SCREEN$ (y-1,x)=32 THEN LET y=y-
1: PRINT AT y,x;a$(n);AT y+1,x;"
"
 691 IF INKEY$="a" THEN IF CODE
SCREEN$ (y+1,x)=32 THEN LET y=y+
1: PRINT AT y,x;a$(n);AT y-1,x;"
"
 692 IF INKEY$="p" AND x<31 THEN
 IF CODE SCREEN$ (y,x+1)=32 THEN
 LET x=x+1: PRINT AT y,x;a$(n);A
T y,x-1;" "
 693 IF INKEY$="o" AND x>0 THEN
IF CODE SCREEN$ (y,x-1)=32 THEN
LET x=x-1: PRINT AT y,x;a$(n);AT
 y,x+1;" "
 694 IF a$(n)="0" OR a$(n)="0" T
HEN GO TO 708
 700 IF x1>x THEN IF CODE SCREEN
$ (y1,x1-1)=32 THEN LET x1=x1-1:
 PRINT AT y1,x1; INK 4;"●";AT y1
,x1+1; INK 7;" "
 701 IF x2>x THEN IF CODE SCREEN
$ (y2,x2-1)=32 THEN LET x2=x2-1:
 PRINT AT y2,x2; INK 4;"●";AT y2
,x2+1; INK 7;" "
 702 IF x1<x THEN IF CODE SCREEN
$ (y1,x1+1)=32 THEN LET x1=x1+1:
 PRINT AT y1,x1; INK 4;"●";AT y1
,x1-1; INK 7;" "
 703 IF x2<x THEN IF CODE SCREEN
$ (y2,x2+1)=32 THEN LET x2=x2+1:
 PRINT AT y2,x2; INK 4;"●";AT y2
,x2-1; INK 7;" "
 704 IF y>y1 THEN IF CODE SCREEN
$ (y1+1,x1)=32 THEN LET y1=y1+1:
 PRINT INK 4;AT y1,x1;"●"; INK 7
;AT y1-1,x1;" "
 705 IF y>y2 THEN IF CODE SCREEN
$ (y2+1,x2)=32 THEN LET y2=y2+1:
 PRINT INK 4;AT y2,x2;"●"; INK 7
;AT y2-1,x2;" "
 706 IF y<y1 THEN IF CODE SCREEN
$ (y1-1,x1)=32 THEN LET y1=y1-1:
 PRINT INK 4;AT y1,x1;"●"; INK 7
;AT y1+1,x1;" "
 707 IF y<y2 THEN IF CODE SCREEN
$ (y2-1,x2)=32 THEN LET y2=y2-1:
 PRINT INK 4;AT y2,x2;"●"; INK 7
;AT y2+1,x2;" "
 708 PRINT AT y,x;a$(n): LET n=n
+1: IF n=5 THEN LET n=1
 709 IF ATTR (y+1,x)=68 OR ATTR
(y-1,x)=68 OR ATTR (y,x+1)=68 OR
 ATTR (y,x-1)=68 THEN LET nivel=
600: GO TO 8900
 710 IF ATTR (y+1,x)=69 OR ATTR
(y,x+1)=69 OR ATTR (y,x-1)=69 TH
EN IF x<21 THEN LET punt=punt+10
0: PRINT INK 5;AT 0,7;punt: PRIN
T INK 7;AT 15,19;" ";AT 16,19;"
": IF CODE SCREEN$ (13,27)=32 TH
EN PRINT INK 6;AT 13,31;"●": FOR
 f=1 TO 30: NEXT f: PRINT AT 13,
31;" "
 711 IF ATTR (y+1,x)=69 OR ATTR
(y,x+1)=69 OR ATTR (y,x-1)=69 TH
EN IF x>21 THEN LET punt=punt+10
0: PRINT INK 5;AT 0,7;punt: PRIN
T INK 7;AT 13,27;" ";AT 14,27;"
": IF CODE SCREEN$ (15,19)=32 TH
EN PRINT INK 6;AT 13,31;"●": FOR
 f=1 TO 30: NEXT f: PRINT AT 13,
31;" "
 715 IF x=31 THEN LET pant=3: LE
T nivel=800: GO TO 8000
 720 GO TO 690
 800 REM ** NIVEL 4 **
 810 CLS : PRINT INK 5;AT 0,0;"P
UNTOS:";punt; INK 6;AT 0,12;"VID
AS:";vid; INK 4;AT 0,21;"RECORD:
";rec
 820 PRINT AT 21,0; INK 5;"■■";
INK 4;"▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼
▼▼▼▼"
 821 PRINT AT 17,2; INK 4;"▼▼▼▼▼
▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼"; INK 5;
"■■"
 822 PRINT AT 13,0; INK 5;"■■";
INK 4;"▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼
▼▼▼▼"
 823 PRINT AT 9,2; INK 4;"▼▼▼▼▼▼
▼▼▼▼"; INK 5;"■■"
 824 PRINT AT 5,14; INK 4;"▼▼▼▼▼"
; INK 3;AT 4,17;"■";AT 3,17;"●"
 825 LET nivel=800: LET ink=1: L
ET x=31: LET y=20: LET mov=1: LE
T a$="O0|0": LET n=1
 850 IF INKEY$="p" AND x<31 THEN
 LET x=x+1: PRINT AT y,x;a$(n);A
T y,x-1;" "
 860 IF INKEY$="o" AND x>0 THEN
LET x=x-1: PRINT AT y,x;a$(n);AT
 y,x+1;" "
 870 IF mov=1 THEN PRINT AT 20,1
5;"●";AT 19,15;" ";AT 20,16;" ";
AT 19,16;"●";AT 16,15;"●";AT 15,
15;" ";AT 16,16;" ";AT 15,16;"●"
;AT 12,15;"●";AT 11,15;" ";AT 12
,16;" ";AT 11,16;"●": LET mov=0:
 GO TO 890
 875 PRINT INK ink;AT 4,17;"■";A
T 3,17;"●"
 876 LET ink=ink+1: IF ink=8 THE
N LET ink=1
 880 IF mov=0 THEN PRINT AT 20,1
5;" ";AT 19,15;"●";AT 20,16;"●";
AT 19,16;" ";AT 16,15;" ";AT 15,
15;"●";AT 16,16;"●";AT 15,16;" "
;AT 12,15;" ";AT 11,15;"●";AT 12
,16;"●";AT 11,16;" ": LET mov=1:
 GO TO 890
 890 IF mov=0 AND x=15 AND y>7 T
HEN GO TO 8900
 891 IF mov=1 AND x=16 AND y>7 T
HEN GO TO 8900
 895 LET n=n+1: IF n=5 THEN LET
```

JAVIER IGUAL

```
n=1
 900 IF ATTR (y+1,x)=69 THEN IF
x=0 OR x=30 OR x=12 THEN FOR f=y
 TO y-3 STEP -1: PRINT AT f,x; I
NK 5;"■■"; INK 7;AT f-1,x;a$(n);
AT f+1,x;"  ": LET y=y-1: NEXT f
: PRINT INK 2;AT f+1,x;"■■"
 910 IF y=4 AND ATTR (y,x+1)<>71
 THEN GO TO 7000
1000 PRINT AT y,x;a$(n): GO TO 8
50
6000 CLS : PRINT AT 0,0;"LO SIEN
TO. NO HAS RECUPERADO A  TU AMO.
 OTRA VEZ SERA."
6010 PRINT AT 21,0;"●": FOR f=2
TO 31: PRINT INK 6;AT 20,f;"■";A
T 21,f;"■";AT 20,f-1;" ";AT 21,f
-1;" ": BEEP .005,10: BEEP .005,
30: BEEP .005,20: NEXT f
6100 PAUSE 200: CLS : IF punt>re
c THEN LET rec=punt: CLS : PRINT
 AT 0,0;"HAS SUPERADO EL ANTERIO
R RECORD": FOR f=1 TO 40: BEEP .
1,f: NEXT f: PRINT AT 10,8;"EL R
ECORD ES ";rec: PAUSE 200: CLS :
 GO TO 8
6110 CLS : GO TO 8
7000 LET punt=punt+1000: PAUSE 2
0: CLS : BEEP .1,10: BEEP .1,11:
 BEEP .5,20: BEEP .1,20: BEEP .1
,21: BEEP .5,30
7010 PLOT 0,87: DRAW 255,0
7020 PRINT INK 4;AT 10,5;"■ ■ ■
■ ■ ■ ■ ■ ■ ■ ■ ■"
7030 FOR f=0 TO 10: FOR n=0 TO 7
: PRINT INK n;AT 21,10;"LO LOGRA
STE!!": NEXT n: NEXT f
7040 FOR f=1 TO 30 STEP 2: PRINT
 AT 10,f;"●": PAUSE 2: PRINT AT
9,f+1;"●";AT 10,f;" ": PAUSE 2:
PRINT AT 10,f+2;"●";AT 9,f+1;" "
: BEEP .005,10: NEXT f: PRINT AT
 10,31;" "
7050 BEEP .1,10: BEEP .2,10: BEE
P .5,15
7060 CLS : PLOT 100,0: DRAW 0,60
: DRAW 50,0: DRAW 0,-60: DRAW -5
0,0: DRAW 0,61: INK 6: DRAW -5,0
: DRAW 0,5: DRAW 60,0: DRAW 0,-5
: DRAW -60,0
7065 DRAW 0,5: DRAW 5,0: INK 7:
DRAW 0,50: DRAW 50,0,-PI: DRAW 0
,-50: CIRCLE 110,110,8: CIRCLE 1
10,112,3: CIRCLE 138,110,8: CIRC
LE 138,112,3
7066 CIRCLE 124,95,5: PLOT 105,8
5: DRAW 40,0,PI/2
7070 PRINT AT 0,0;"QUE BIEN!! HA
S ENCONTRADO TU AMO"
7080 BEEP .1,15: BEEP .1,10: BEE
P .1,10: BEEP .1,15: BEEP .1,20:
 BEEP .1,15: BEEP .1,15: BEEP .1
,20
7090 BEEP .1,15: BEEP .1,10: BEE
P .1,10: BEEP .1,15: BEEP .1,20:
 BEEP .1,15: BEEP .1,15: BEEP .1
,20: BEEP 1,0
7100 CLS : CIRCLE INK 6;128,88,5
0: CIRCLE INK 6;128,88,30
7110 PRINT INK 3;AT 0,0;"QUE PAS
A? TU AMO SE HA PERDIDO  OTRA VE
Z. TIENES QUE RECUPERARLO"
7120 PRINT INK 4;AT 19,0;"TIENES
 QUE VOLVER A LA GRUTA    DENTRO
 DE"; INK 7;AT 21,15;"20 SEGUNDO
S"
7130 LET sec=20
7140 IF sec=-1 THEN BEEP 1,10: G
O TO 100
7150 PRINT AT 21,15;sec;" ": LET
 sec=sec-1: PAUSE 50: GO TO 7140
8000 CLS : PAUSE 10: PRINT INK 1
;AT 10,10;"F          S"; INK 2;A
T 10,11;"E         E"; INK 3;AT 10
,12;"L      D"; INK 4;AT 10,13;"I
   A"; INK 5;AT 10,14;"C D"; INK
 6;AT 10,15;"I"
8001 PRINT AT 0,0;"PASASTE LA GR
UTA <<";pant;">> !!!!...."
8002 BEEP .3,10: BEEP .3,15: BEE
P .3,10: BEEP .5,20: LET a$="PUL
SA UNA TECLA PARA CONTINUAR..":
FOR n=1 TO 32: PRINT #1;a$(n);:
BEEP .01,n: NEXT n
8003 IF INKEY$="" THEN GO TO 800
3
8004 GO TO nivel
8900 CLS : LET vid=vid-1: IF vid
=0 THEN FOR f=40 TO 0 STEP -1: B
EEP .01,f: NEXT f: PRINT AT 10,1
0;"SE ACABO!!": BEEP 2,-20: FOR
f=1 TO 100: NEXT f: GO TO 6000
8950 FOR f=0 TO 20: BEEP .01,f:
NEXT f: FOR f=20 TO 0 STEP -1: B
EEP .01,f: NEXT f: PRINT AT 10,8
;"VIDAS:  ";: FOR f=1 TO vid: PR
INT "● ";: NEXT f: PRINT AT 13,8
;"PUNTOS:  ";punt: FOR f=0 TO 10
0: NEXT f: GO TO nivel
9900 FOR f=0 TO 7: READ a: POKE
USR "a"+f,a: NEXT f
9910 DATA 60,126,BIN 11100111,BI
N 11000011,BIN 11000011,BIN 1110
0111,126,60
9920 FOR f=0 TO 7: READ a: POKE
USR "b"+f,a: NEXT f
9930 DATA BIN 00011100,BIN 00111
110,BIN 00100110,BIN 00100110,BI
N 00100110,BIN 00100110,BIN 0011
1110,BIN 00011100
9940 FOR f=0 TO 7: POKE USR "c"+
f,BIN 00011000: NEXT f
9950 FOR f=0 TO 7: READ a: POKE
USR "d"+f,a: NEXT f
9960 DATA BIN 00111000,BIN 01111
100,BIN 01100100,BIN 01100100,BI
N 01100100,BIN 01100100,BIN 0111
1100,BIN 00111000
9961 FOR f=0 TO 7: READ a: POKE
USR "e"+f,a: NEXT f: DATA 60,126
,BIN 11011011,255,BIN 10111101,B
IN 11000011,255,255
9962 FOR f=0 TO 7: READ a: POKE
USR "f"+f,a: NEXT f: DATA 60,126
,255,255,255,255,255,0
9963 FOR f=0 TO 7: READ a: POKE
USR "g"+f,a: NEXT f: DATA 0,255,
255,255,255,255,255,255
9964 FOR f=0 TO 7: READ a: POKE
USR "h"+f,a: NEXT f: DATA 255,25
5,BIN 01111110,BIN 00111100,BIN
00011000,0,0,0
9965 FOR f=0 TO 7: READ a: POKE
USR "i"+f,a: NEXT f: DATA BIN 10
000001,BIN 10000001,BIN 10000001
,255,BIN 10000001,BIN 10000001,B
IN 10000001,255
9966 FOR f=0 TO 7: READ a: POKE
USR "j"+f,a: NEXT f: DATA 60,BIN
 01011110,BIN 10111111,BIN 10111
111,BIN 10111111,BIN 10111111,BI
N 01011110,60
9970 FOR f=0 TO 7: READ a: POKE
USR "k"+f,a: NEXT f
9971 DATA BIN 01111110,255,255,2
55,255,255,255,BIN 01111110
9975 FOR f=0 TO 7: READ a: POKE
USR "l"+f,a: NEXT f: DATA 0,0,0,
BIN 00011000,BIN 00111100,BIN 01
111110,255,255
9980 FOR f=0 TO 7: READ a: POKE
USR "m"+f,a: NEXT f: DATA 60,126
,BIN 11011011,255,BIN 11000011,B
IN 10111101,255,255
9999 RETURN
```

¡NUEVO!

La ciudad de las Sombras

NIGHTSHADE

Ultimate
Videoaventura
Importado



De repente, sin que nadie lo esperara y olvidándose de aquellas largas campañas publicitarias —a las que nos tenía acostumbrados antes de que saliera el producto— Ultimate acaba de lanzar su última obra. Y como ocurre en el mundo del espectáculo, nos hemos apresurado a asistir al estreno.

Antes de abrirse el telón vemos la típica presentación de rigor y, como todas las demás, ésta también se encuentra rodeada de un halo de mágico misterio que lo envuelve todo. Una casa medieval se muestra impávida ante nuestros ojos, entre sombras, con extraños símbolos de brujería colgados sobre las ventanas. Estamos en la larga Noche de las Sombras.

Se abre el telón y tras la melodía orquestada que nos anuncia el comienzo de la primera escena, aparece el protagonista. Allí está, solo en el centro de la calle, mirando a su alrededor. De repente, un desfile de extraños seres empieza a pulular alrededor nuestro: pequeños monstruos cabezudos, llamas con vida propia, una especie de luces destellantes y figuras fantasmagóricas, componen la nutrida fauna nocturna.

El objetivo, acabar con los enemigos de la humanidad antes de que sea demasiado tarde. Son cuatro y están representados mediante unas figuras situadas en la parte inferior derecha de la pantalla. Enseguida los conoceremos, su presencia domina la ciudad. Están ahí detrás de cualquier esquina, o en el interior de una de las mansiones de la ciudad. Nosotros tenemos que encontrarlos y destruirlos, pero la tarea no es fácil. No son seres que se les pueda derrotar de cualquier modo, sólo existe un arma a la que son vulnerables, y por supuesto, sólo sirve para acabar con uno de ellos, pero como hay cuatro, tenemos que averiguar cuál es la que afecta a cada personaje.

Ocurre curiosamente que cuando recogemos algunos de estos objetos parpadearán en el momento que se encuentre cerca el enemigo, al que puede destruir.

A diferencia de otras producciones de Ultimate aquí nuestro personaje puede disparar a sus enemigos. Para ello hay repartidas por la ciudad una serie de armas de las que es posible disponer en cualquier momento. Se pueden usar como defensa contra la corte de monstruos que nos acechan, pero nunca contra los cuatro enemigos principales, porque como ya hemos dicho, éstos sólo son vulnerables a un tipo de objeto.

El número de vidas se encuentra señalizado mediante unos muñecos que van cambiando de color, al igual que el protagonista de nuestro juego que va tomando diferentes tonos según se va debilitando su energía. También puede recuperarla recogiendo una pócima vitalizadora que encontrará en algunos lugares de la ciudad.

Si después de todo conseguimos completar la arriesgada misión, asistiremos al final de las fuerzas maléficas de la noche que desaparecerán para siempre en lo más profundo de las entrañas de la Ciudad de las Sombras. Entonces, al igual que en los cuentos de hadas, volverá a lucir el sol.







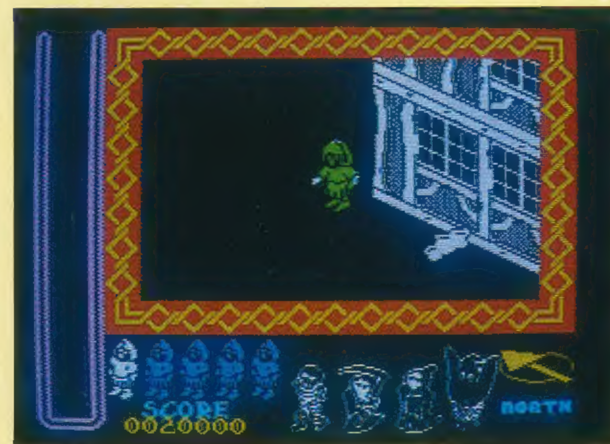






¡NUEVO!









RESUMIENDO...

Originalidad. Se ha cambiado el tema, se ha modificado el tratamiento gráfico, se han creado nuevos personajes, pero no se ha conseguido que el producto sea original. Sigue dentro de la misma línea de juegos anteriores.

* *

Gráficos. Aquí, una vez más, Ultimate sigue marcando la pauta. En esta ocasión ha cambiado la técnica de la perspectiva, ahora son los objetos los que se mueven en relación al personaje central. Además utiliza tres colores.

* * * * *

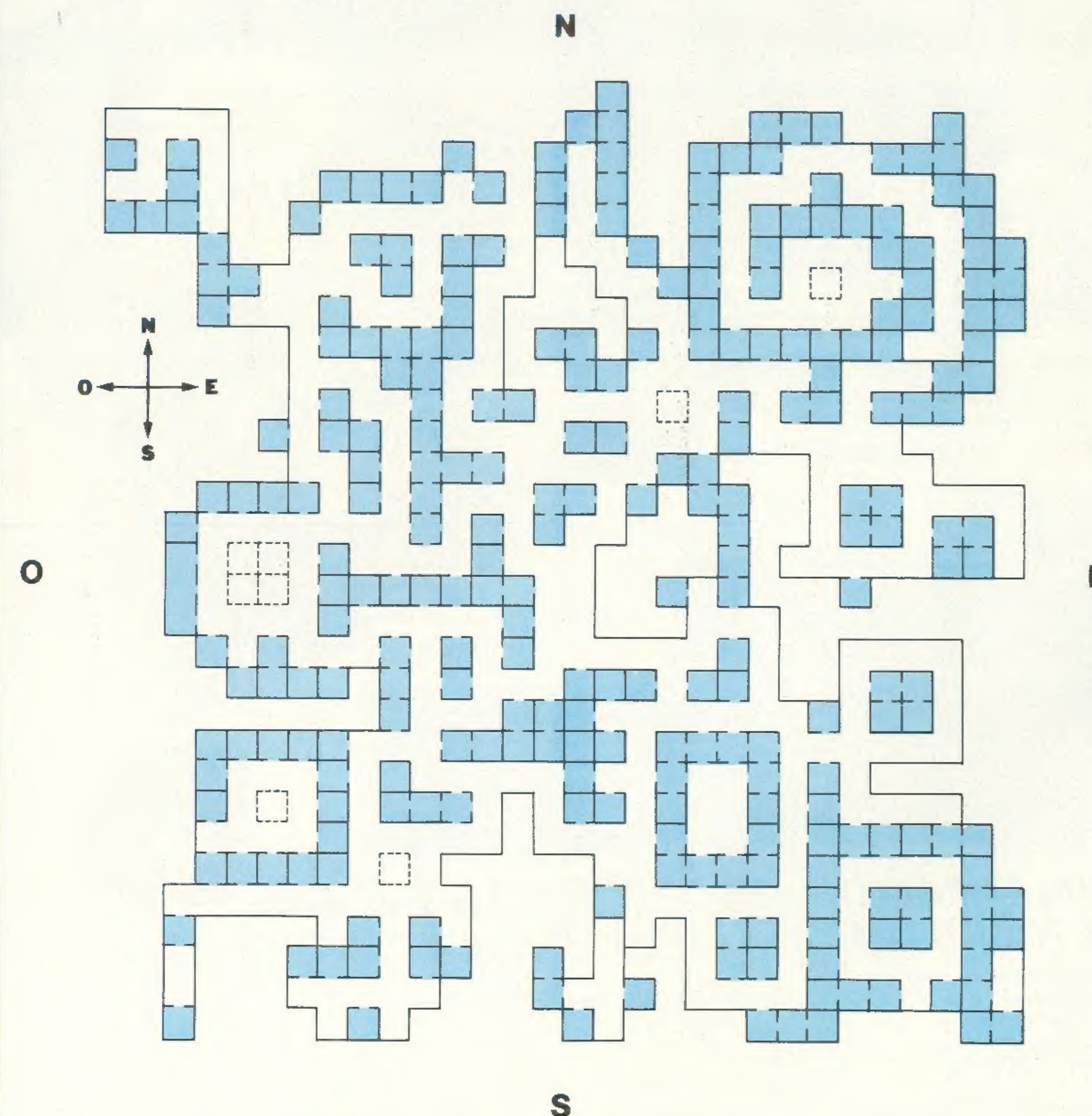
Sonido. Las melodías que ya vienen siendo costumbre y algunos efectos sonoros.

* * *

Valoración. Muy bien, con muchos detalles de buen gusto y con bastantes mejoras con respecto a otros programas. Sin embargo, hay que reconocer que Ultimate debe cambiar y pronto, antes de que se le agote la inspiración.

* * * *

MAPA DEL NIGHTSHADE



CODIGO OPERACION	
CODIGO OPERACION	
Ø Ø 1 1 1 Ø 1 1	3Bh
Ø 1 1 1 1 1 1 1	7Fh

Existe una combinación diferente de bits para cada instrucción en el código de operación, a pesar de que la única diferencia sea el registro usado. Por lo tanto, habrá una parte fija y otra variable en el código de operación en función de dichos registros.

Otra cosa muy curiosa es ver cómo el microprocesador va leyendo una serie de octetos llenos. de instrucciones cuando estos pueden tener cuatro longitudes distintas. Es muy sencillo. Se recordará que el microprocesador tiene un registro PC que indica la dirección de la memoria donde está la próxima instrucción a ejecutar. Cuando la lee, en función del código de operación, sabe cuantos octetos ocupa; en ese momento sólo tiene que incrementar el registro PC en esa cantidad, mientras va leyendo los restantes octetos de la instrucción, para que cuando vaya a leer la siguiente instrucción, este registro ya la tenga apuntada. Ver FIGURA 4-2.

Necesidad de conocer el código máquina

Podríamos preguntarnos: voy a escribir mi programa en el lenguaje simbólico ASSEMBLER, tengo un ensamblador magnífico que me lo va a traducir, ¿qué necesidad tengo de conocer que bit tiene que estar activo y cual no?

Desde luego, el que tenga un buen ensamblador y escriba un programa de principio a final, buena gana de estar descifrando códigos y pasándolos de binario a hexadecimal. Pero no siempre se dispone de un buen ensamblador o se quiere estar cargándolo para pocas instrucciones, con lo cual se meterían en la memoria los valores de las instrucciones y se saltaría a esa posición con lo que el microprocesador empezaría a trabajar. Además, existen algunas técnicas en programación que hacen necesario este conocimiento, por ejemplo:

1. Programas automodificables: La primera vez que pasan por una rutina pasan por una instrucción que ejecuta una operación; posteriormente, esa instrucción se modifica con lo que la próxima vez que pase ejecuta una operación distinta.

2. Parches: Durante la prueba del programa se encuentra un error; conociendo las posiciones de memoria donde está cargado el programa se puede modificar directamente y seguir la prueba, incluso si la modificación ocupa más espacio, se deja una zona del programa definida a ceros binarios, se hace sobre ella la modificación terminando con un retorno y desde la posición de error, se hace una llamada a esa ru tina.

Por otro lado, no es necesario tener un ensamblador para poder utilizar el Spectrum en código máquina; es posible escribir el programa en Assembler y traducirlo a código máquina, manualmente; si el programa no es muy largo, la labor no resulta excesivamente tediosa. De hecho, es preferible aprender primero a ensamblar «a mano», ya que esto proporciona un conocimiento más profundo del Assembler, y facilita la utilización posterior de un programa ensamblador.

En este mismo curso, describimos en detalle la utilización del que consideramos el mejor ensamblador que se ha escrito para el Spectrum, el "GENS 3", que desde ahora, recomendamos a nuestros lectores; mientras tanto, indicaremos con todo detalle la forma de ensamblar "a mano" cada instrucción y todos los ejemplos que demos, se podrán introducir en el ordenador sin necesidad de ensamblador, por lo que para seguir este curso, sólo es necesario disponer de un Spectrum.

Formatos de instrucción en lenguaje simbólico

Como se recordará, el lenguaje simbólico es aquél en el que escribimos el programa fuente. Los códigos nemotécnicos que se han utilizado para el microprocesador Z-8Ø, son abreviaturas de las palabras inglesas que definen la operación que realizan.

El formato de la instrucción y sus normas de sintaxis pueden variar algo en función del ensamblador elegido; pero siempre hay unas reglas mínimas que suelen cumplir y a esas nos referiremos hasta el capítulo que trate más profundamente el ensamblador.

El formato normal es el siguiente:

ETIQUETA NEMOTECNICO OPERANDOS ; COMENTARIOS

En el Spectrum se ha utilizado una versión más moderna denominada Z-8ØA, que admite señales de reloj de hasta 4 MHz, con lo que se consigue una mayor velocidad de ejecución.

En el Spectrum se ha utilizado una señal de reloj de 3.5 MHz en vez de los 4 tolerados, para evitar llevar al microprocesador al límite de su frecuencia de trabajo, lo que podría dar lugar a errores.

CICLOS DE MAQUINA Y CICLOS DE INSTRUCCION

Se denomina *Ciclo de instrucción* al tiempo durante el cual el microprocesador ejecuta una instrucción completa.

El ciclo de instrucción se subdivide a su vez, en ciclos de máquina. Un ciclo de máquina es el tiempo durante el cual el microprocesador realiza una operación elemental. Cada ciclo de máquina emplea varios ciclos (impulsos) de reloj.

Se denomina "M1" al ciclo de máquina correspondiente a la búsqueda del código de operación, durante el cual, la pata M1 del microprocesador se coloca a nivel bajo. El ciclo de máquina M1 ocupa 4 ciclos de reloj; un ciclo de reloj dura aproximadamente Ø.29 microsegundos (millonésimas de segundo), por lo que el ciclo M1 dura 1.14 microsegundos.

Un ciclo de memoria es una operación de lectura o escritura en memoria, emplea 3 ciclos de reloj, y dura Ø.86 microsegundos.

En la FIGURA 6 se puede apreciar el cronograma (diagrama de tiempos) de una instrucción típica.

INSTRUCCIONES DE UN SOLO OCTETO

1er Byte — CODIGO DE OPERACION

INSTRUCCIONES DE DOS OCTETOS

1er Byte — CODIGO DE OPERACION
2º Byte — DATOS O DIRECCIONES

INSTRUCCIONES DE TRES OCTETOS

1er Byte — CODIGO DE OPERACION
2º Byte — DATOS O DIRECCIONES
3er Byte — DATOS O DIRECCIONES

INSTRUCCIONES DE CUATRO OCTETOS

1er Byte — CODIGO DE OPERACION
2º Byte — CODIGO DE OPERACION
3er Byte — DATOS O DIRECCIONES
4º Byte — DATOS O DIRECCIONES

Figura 5. Formatos de instrucción del Z-80.

CICLO DE RELOJ

SEÑAL DEL RELOJ

T1 T2 T3 T4 T1 T2 T3 T1 T2 T3

M1 — BUSQUEDA DEL CODIGO DE OPERACION

M2 — LECTURA DE MEMORIA

M3 — ESTRUCTURA EN MEMORIA

CICLO DE INSTRUCCION

Figura 6. Cronograma de un ciclo típico de instrucción.

TIEMPOS DE EJECUCION

Como el lector habrá deducido ya, es posible calcular el tiempo de ejecución de una determinada rutina en C/M, a condición de conocer el número de ciclos de reloj que emplea cada una de sus instrucciones.

En lo sucesivo, cada vez que veamos una determinada instrucción, indicaremos el número de ciclos de reloj que emplea el microprocesador para ejecutarla, así como el número de veces que accede a memoria (ciclos de memoria).

Como ejemplo, veamos lo que se tarda en cargar el registro "A" con un número. Podemos utilizar la instrucción: LD A, # FF que carga el número 255 (FFh) en el acumulador; esta instrucción accede 2 veces a memoria (2 ciclos de memoria), una para buscar el código de operación (4 ciclos de reloj) y otra para buscar el número que ha de cargar en "A" (3 ciclos de reloj); lo que hace un total de 7 ciclos de reloj, es decir, unos 7 x 0.29 = 2 microsegundos. Este ejemplo ilustra la enorme velocidad del código máquina, el microprocesador es capaz de cargar el acumulador medio millón de veces en un segundo.

Supongamos que queremos sumar en el acumulador una lista de números, y usamos el registro HL para movernos a lo largo de esa lista, el bucle podría ser:

```
BUCLE ADC A, (HL)
      INC HL
      JR BUCLE
```

El tiempo de ejecución por cada pasada, sería de 25 ciclos de reloj, es decir 7.14 microsegundos. Con este bucle, nuestro ordenador podría sumar 140000 números por segundo. Por supuesto, este pequeño bucle no es operativo, no existe condición de salida del bucle, por lo que el ordenador se quedaría eternamente atrapado dentro de él, y por otro lado, se produciría un rebosamiento en el acumulador, ya que el resultado de las sumas excedería su capacidad, a menos que la mayor parte de los números fueran ceros. La única finalidad de este ejemplo es mostrar la enorme velocidad de ejecución del código máquina.

Para calcular el tiempo de ejecución de sus rutinas, sume los números de ciclos de reloj de cada instrucción y multiplique por 0.29 para obtener el resultado aproximado en microsegundos.

Modos de direccionamiento

En la mayor parte de las operaciones, el Z-80 utiliza datos almacenados en sus registros o en posiciones de memoria. Las formas posibles de indicarle la situación de estos datos, constituyen los diversos modos de direccionamiento.

DIRECCIONAMIENTO INMEDIATO:

En este modo de direccionamiento, el byte que sigue al código de operación en memoria, contiene el operando.

CODIGO DE OPERACION

uno o dos bytes

OPERANDO

un byte

Un ejemplo podría ser cargar el acumulador con una constante, donde la constante es el byte que sigue al código de operación.

DIRECCIONAMIENTO INMEDIATO EXTENDIDO:

Es igual que el anterior, salvo que el operando ocupa dos bites, el primer byte es el octeto de orden bajo, y el segundo, el de orden alto.

CODIGO DE OPERACION

uno o dos bytes

OPERANDO (LSB)

Octeto de menos peso

OPERANDO (MSB)

Octeto de más peso

Un ejemplo podría ser la carga de un registro doble con una constante que, lógicamente, ocuparía dos bytes de memoria.

DIRECCIONAMIENTO RAPIDO DE PAGINA CERO

El Z-80 tiene unas instrucciones de salto rápido a una dirección de página cero. Hay ocho direcciones posibles donde se colocan las rutinas de uso más frecuente, de esta forma se puede llamar a estas rutinas empleando un solo byte.

En el Spectrum, estas direcciones se encuentran utilizadas por la ROM para las rutinas de más uso, y son las siguientes:

```
RST 00h : Inicialización
RST 08h : Salida de error
RST 10h : Imprimir un caracter
RST 18h : Leer un caracter
RST 20h : Leer el siguiente
          caracter
RST 28h : Entrada al calculador
RST 30h : Hacer espacio en
          memoria
RST 38H : Leer el teclado
```

DIRECCIONAMIENTO RELATIVO

En este caso, el byte que sigue al código de operación se emplea como un entero en complemento a dos, que se suma a la dirección actual almacenada en el «PC».

CODIGO DE OPERACION

un byte (salto relativo)

OPERANDO

entero en complemento a 2

Este modo de direccionamiento permite efectuar saltos relativos, con lo que las rutinas pueden ser *reubicables* es decir, correr de igual forma en cualquier dirección de memoria.

DIRECCIONAMIENTO INDEXADO

En esta forma de direccionamiento, el byte de datos que sigue al código de operación contiene un entero de desplazamiento en complemento a dos, que se suma al contenido actual del registro índice correspondiente, para apuntar a una dirección de memoria. El código de operación tiene siempre dos bytes, el primer byte es «DDh» siempre que se utilice el registro «IX» y «FDh» siempre que se utilice el «IY».

DD o FD

índice usado

CODIGO DE OPERACION

un byte

DESPLAZAMIENTO

entero en compl. a dos

DIRECCIONAMIENTO DE REGISTROS

En muchos de los códigos de operación, hay ciertos bits que especifican a qué registro se refiere la instrucción, permaneciendo inalterados el resto de los bits. Un ejemplo podría ser la instrucción:

```
LD C,B
```

Que significa: «cargar en el registro "C" el contenido del registro "B"».

DIRECCIONAMIENTO IMPLICITO

En este caso, la situación de los datos está implícita en el código de operación. Por ejemplo, en las operaciones aritméticas de 8 bits, el registro «A» (acumulador) es siempre el que recibe los resultados.

DIRECCIONAMIENTO INDIRECTO

En esta forma de direccionamiento, el contenido de un registro doble se utiliza como dirección a partir de la cual hay que cargar el dato. Un ejemplo podría ser:

```
LD A, (HL)
```

Que significa: «carga el registro A con el contenido de la dirección de memoria apuntada por el registro HL». En este caso el registro HL se utiliza como *puntero* para «apuntar» a una dirección de

PROGRAMACION EN ASSEMBLER

En este capítulo se está ya en condiciones de saber qué es la programación en ASSEMBLER; esto es, escribir una serie de códigos entendibles por el usuario que posteriormente serán convertidos en código de máquina entendible por el microprocesador, en este caso el Z-80.

La programación en ASSEMBLER requiere cuidados especiales si se desea sacar el máximo rendimiento, por ejemplo, ante dos instrucciones que obtengan el mismo resultado se debe elegir aquélla que tenga menos ciclos de máquina o de reloj, o aquélla que ocupe menos posiciones de memoria; incluso en algunos casos habrá que elegir entre ocupar menos posiciones o ser más rápidos, en función de las necesidades que se tengan. Esto no quiere decir que sea necesario conocer de memoria los ciclos de cada instrucción; un manual de ASSEMBLER debe contener toda la información necesaria, con un método de acceso fácil, a pesar de que en algún caso resulte redundante. Se pretende que este curso, además de como tal, pueda servir en el futuro como un manual de consulta rápida, por lo que en algunos casos, es posible que el lector encuentre información reiterada.

Otra buena costumbre cuando se programa en ASSEMBLER es poner comentarios; siempre hay una manera de ponerlos en cada instrucción o intercalados entre ellas. Los comentarios sólo ocupan lugar en el código simbólico o programa fuente; cualquier ensamblador los ignora cuando convierte el programa en código de máquina, lo cual quiere decir que no ocupará más un programa absoluto porque su simbólico tenga comentarios, pero tampoco irá más despacio. Cuando pase el tiempo y queramos modificar alguna parte del programa y se haya olvidado el porqué de cada instrucción, los comentarios serán de gran ayuda.

Siguiendo con la exposición de buenas costumbres nos referiremos, por último, al empleo de subrutinas. Ya veremos cómo se hacen y cómo se accede a ellas, pero hay que irse mentalizando a su uso. Esto es importante en este momento porque se trata de un problema de estructura del programa. Las ventajas son múltiples; una estructura de subrutinas es más fácil de entender, por lo tanto de modificar. Se da con frecuencia el caso de necesitar en un programa operaciones iguales o semejantes a las de otro, por lo tanto, con limitarse a copiar totalmente estas partes o como mucho, adaptarlas algo a las características del nuevo programa, saldríamos adelante.

Resumiendo, hay que acostumbrarse desde un principio a estos métodos y a la realización del organigrama, sobre el cual daremos algunas orientaciones al final de este capítulo. Toda esta información, más los diseños que se hagan de pantalla, de campos o de tablas, se guardarán juntas en una carpeta o cosidas con grapa y se conseguirá una buena documentación de cada programa, documentación que nos será muy útil en el futuro.

Realización de un programa

Existen una serie de pasos que se deben seguir para dar por definitiva la realización de un programa. Cuanto más detallada sea la organización de estos pasos, más fácil será la realización del siguiente. El concepto programa abarca a todo lo que se puede realizar en una corriente de instrucciones dentro de un ordenador, por lo cual, tan programa es un juego, como llevar la contabilidad de una casa o la solución de problemas científicos; la metodología a seguir es la misma.

PASOS A SEGUIR:

a) **Planteamiento del programa:** Lo que en la informática profesional se llama *análisis funcional.* Es la definición de lo que se quiere realizar, la información de que se dispone, la información que se quiere obtener, la información que se quiere volver a utilizar, los formatos de pantalla deseados; todas estas definiciones se deben hacer con el máximo detalle, como si se pretendiese explicar a otra persona cuál es nuestro problema y cómo nos gustaría que nos lo solucionase.

b) **Viabilidad de resolver el problema con un ordenador:** Lo que en informática profesional se llama *análisis técnico u orgánico.* En este paso se valorará la posibilidad de resolver el problema con el ordenador de que se dispone, así como de la periferia a nuestro alcance (impresora, microdrive, etc.). Se definirá toda la información del paso anterior como datos entendibles para el ordenador, indicando en qué código irán, cuánta memoria ocuparán, en qué lugar han de almacenarse y cómo se procesarán; lo mismo para los diseños de pantalla. En este paso es donde se debe decidir qué procesos se hacen por medio de subrutinas, y, como curiosidad, si el programa se hace en BASIC o en ASSEMBLER.

c) **Realización del programa:** Es decir, la programación propiamente dicha. Este paso se divide en dos partes: la construcción del organigrama y la codificación. Estos dos pasos son complementarios, cuanto más se detalle el organigrama menos vueltas se da a la codificación, y viceversa.

1. El organigrama es una construcción gráfica del flujo o caminos posibles que tiene el programa.

2. La codificación es la escritura del lenguaje simbólico contemplando todas las alternativas posibles definidas en el organigrama.

Por supuesto, hay muchos programadores que nunca hacen un organigrama, pero también hay muchos que se olvidan de codificar una rama. En cualquier caso, para un programa sencillo o bien para un programador experto el organigrama puede no ser muy detallado o bien omitirse.

d) **Prueba del programa:** En este paso se preparan unos datos de entrada al programa de los cuales ya se conocen los resultados que obtienen, contemplando, si es posible, todos los casos. Por ejemplo, si se hace un programa para resolver raíces cuadradas, se introducirán una serie de valores de los que ya se conoce el resultado.

e) **Documentación:** Una vez concluidos los pasos anteriores, se reúne todo el material e incluso se comentan los problemas o dificultades encontrados y cómo se han solucionado. Dependerá de lo ordenado que sea cada cual el obtener como resultado una buena documentación. Ver FIGURA 4-1.

Todo lo anteriormente descrito se puede hacer o no, pero son métodos ya muy estudiados que producen unos buenos resultados. La metodología anteriormente descrita es una orientación de la más elemental; se pueden usar técnicas más complicadas pero que se salen un poco de lo que es un micro-ordenador. Se dice que en un programa vale todo, siempre y cuando funcione. Si existe un método que lleve a un buen resultado, no sólo que funcione, sino que sea lo más rápido posible, que no se repitan procesos, que ocupe la menor cantidad de memoria y que esté claro, ¿por qué no usarlo?

Formatos de instrucción en código máquina

La memoria del SPECTRUM está dividida en octetos. Un octeto es una agrupación de 8 bits y un bit es la unidad más pequeña de información; sólo puede informar de dos estados: o está activo o no. Cuando de un SPECTRUM se dice que tiene 16K indica que tiene, aproximadamente, 16.000 octetos de memoria disponible, que son exactamente en decimal 16.384; para uno de 48K se multiplica esta cantida por 3.

Los 48K de memoria disponible en RAM más los 16K que usa el programa monitor en ROM, suman 64K, que son 65.536 octetos. Esto es, desde la posición 0 de memoria a la 65.535, que es en hexadecimal FFFFh y en binario 1111111111111111b y a su vez la máxima cantidad que se puede escribir en dos octetos.

Las intrucciones en el SPECTRUM pueden ocupar 1, 2, 3 ó 4 octetos en función del código de operación y básicamente, tiene los formatos que se ven en la FIGURA 5 del CAPITULO 3.

Como se vé, lo que es común a todos los formatos es que los primeros tienen el código de operación y los últimos el operando. Sobre el operando hay que tener en cuenta que cuando son dos octetos el microprocesador espera encontrar el octeto de orden inferior en el primer octeto del operando, y el de orden superior en el segundo, de tal forma que al referirse al valor de operando 7F3Bh el ensamblador almacenará la instrucción en memoria de la siguiente manera:

memoria, siempre que un registro se utilice como puntero, su nombre aparecerá, en el código simbólico, encerrado entre paréntesis, significando: «donde apunta el contenido de».

También se puede utilizar como puntero, una constante de dos bytes, ensamblada a continuación del código objeto, por ejemplo:

```
LD A, (5C37)
```

Que significa: «carga el registro A con el contenido de la dirección de memoria 5C37h". Cuando hagamos estos en Assembler, normalmente utilizaremos una «etiqueta» de la siguiente forma:

```
LD A, (ETIQUE)
ETIQUE EQU #5C37
```

De esta forma, solo tendremos que definir la etiqueta una vez, pero podremos usarla todas la veces que queramos sin tener que recordar de memoria los números. Los nombres de las variables del Sistema en el Spectrum son, precisamente, etiquetas del código fuente del Sistema Operativo. El uso de las etiquetas se verá en profundidad cuando estudiemos el manejo de ensambladores.

En algunos casos, el direccionamiento indirecto se utiliza para especificar operandos de 16 bits (dos bytes), en este caso, el puntero apunta al byte de menos peso, siendo el de más peso el siguiente. Por ejemplo:

```
LD HL, (5C37)
```

Que significa: «carga el registro L con el contenido de la posición de memoria 5C37h, y el registro H con el contenido de la posición de memoria siguiente (5C38h)».

DIRECCIONAMIENTO DE BITS

Un gran número de instrucciones del Z-8Ø trabajan directamente sobre bits individuales de registros o posiciones de memoria. En este caso, se utiliza uno de los métodos de direccionamiento anteriores para indicar el registro o posición de memoria en concreto, y tres bits del código de operación para indicar a qué bit de ese registro o posición de memoria nos referimos.

MODOS DE DIRECCIONAMIENTO COMBINADOS

Muchas instrucciones incluyen más de un operando, en estos caso, se pueden combinar más de un modo de direccionamiento dentro de una misma instrucción, por ejemplo:

```
LD (IX+7),A
```

Que utiliza directamente indexado para el destino y direccionamiento inmediato para la fuente. Significa: «carga en la posición de memoria apuntada por el contenido del registro IX más 7, el contenido del registro A (acumulador)».

Instrucciones del Z-8Ø

El Z-8Ø puede ejecutar un gran número de instrucciones, podemos ordenarlas en los siguientes grupos:

CARGA E INTERCAMBIO

Permiten desplazar datos entre registros, o entre estos y posiciones de memoria. También se puede intercambiar el contenido de dos registros, o el de dos grupos alternativos.

ARITMETICAS Y LOGICAS

Permiten realizar operaciones aritméticas o lógicas entre el acumulador y un registro o posición de memoria. Los resultados se almacenan en el acumulador, y los indicadores del registro «F» se ponen a «1» o a «Ø» en función del resultado de la operación.

BUSQUEDA Y TRANSFERENCIA DE BLOQUES

Se trata de las más poderosas instrucciones del Z-8Ø, es posible transferir todo un bloque de mèmoria con una sola instrucción; también es posible examinar todo un bloque de memoria para buscar un determinado dato de un byte.

ROTACION Y DESPLAZAMIENTO

Permiten la rotación bit a bit del dato almacenado en un registro o una posición de memoria, las rotaciones pueden incluir el indicador de acarreo del registro «F».

MANIPULACION DE BITS

Permiten tratar de formar independiente cada bit de un registro o una posición de memoria, es posible poner un bit a «1», ponerlo a «Ø» o examinar si es «1» o «Ø».

SALTO LLAMADA Y RETORNO

Permite alterar la secuencia normal del programa para saltar a otro lugar de la memoria o ejecutar una subrutina. También es posible retornar desde una subrutina al punto donde se la llamó.

ENTRA Y SALIDA

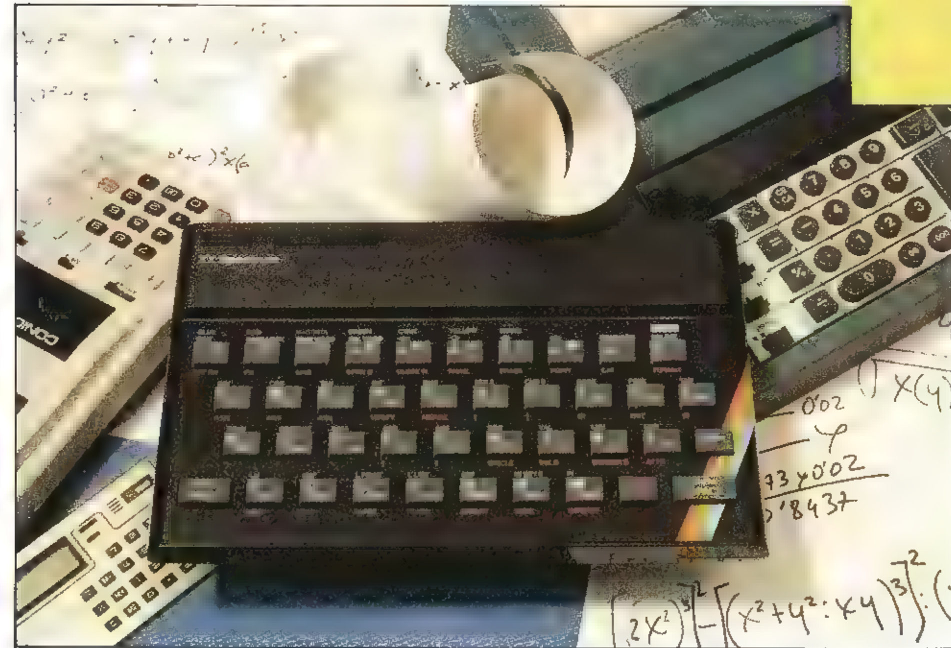
Permiten leer y escribir datos en los ports de entrada/salida, con lo cual se comunica el ordenador con el mundo exterior.

CONTROL CPU

Se utilizan para controlar el propio funcionamiento del microprocesador, inhibir o habilitar interrupciones, cambiar el modo de interrupción, detener el funcionamiento del microprocesador, etc.

En los capitulos posteriores de este curso, se irán viendo detenidamente, una a una, todas las instrucciones de cada uno de estos grupos y la forma de utilizarlas en nuestros programas.

Antes, en el capitulo siguiente, se verán los conceptos básicos de la programación en Assembler, y las formas de almacenar y ejecutar nuestros programas en Código Máquina.



Breves notas sobre cómo exterminar a los Gremlins (Capítulo IV)

por Billy Puertas



Estoy en la puerta del cine COLONY. A juzgar por las huellas que conducen hasta allí, seguramente hay en su interior montones de Gremlins. Me dirijo hacia el cine y luego al Sur. Empiezo a proyectar antes de que esas horribles criaturas invadan la sala.

Voy al Norte hasta encontrar los grandes almacenes y entro en ellos.

Estoy en el hall de los grandes almacenes y por el Este salgo a la escalera. Me dirijo arriba hasta encontrar el escotillón. Suelto la escala (con su ayuda podré subir fácilmente al tejado) y voy hasta ella.

Acabo de subir al tejado de los grandes almacenes. Todo el sitio está nevado y un gran agujero en los sistemas de ventilación me hace temer lo peor.

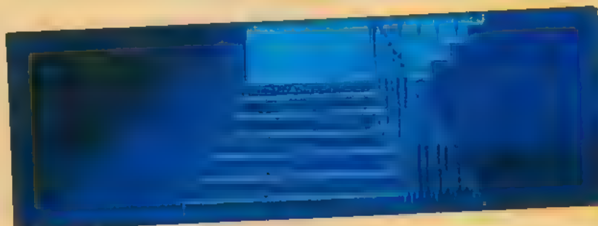
Gizmo, como presintiendo una inminente tragedia, se introduce a toda velocidad por el sistema de ventilación.

Oigo ruidos... ¡Pero que es esto!

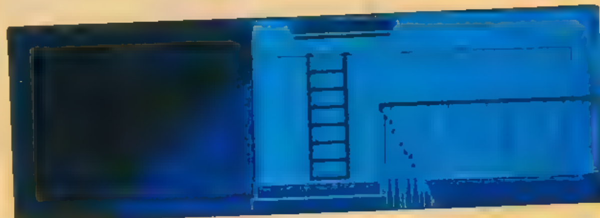
¡Tengo visitas algo desagradables! ¡Los Gremlins están aquí de nuevo! Tras empujar el botón de la cámara fotográfica se produce una luz cegadora y los verdes diablillos huyen aterrorizados por el sistema de ventilación.

Abro la válvula y enciendo el soplete. Ahora he de soldar la chapa para evitar que puedan escapar por el agujero. Lo siento por Gizmo; espero que encuentre otra salida.

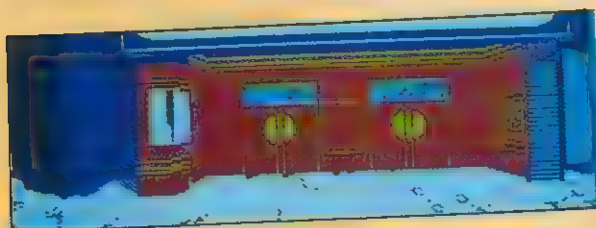
Bajo del tejado y me encuentro de nuevo en la escalera. Yendo hacia el Oeste por el pasillo, me encuentro con una puerta que en principio está cerrada pero... ¡un momento!... ¡parece que se mueve!: Gizmo, sano y salvo, me ha abierto la puerta.



El rellano de la escalera que sube hasta el último piso.



Con el auxilio de la escala resulta fácil acceder al tejado.



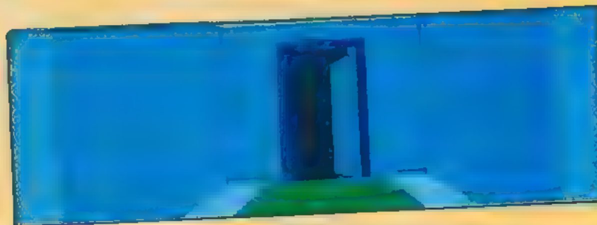
La entrada del Cine Colony, con las pisadas delatoras.



Un enorme agujero evidencia la presencia de los diablillos.



Un pequeño retraso en la película y la sala será invadida.



Mi amigo GIZMO me abre la puerta desde el interior.

Premiado con 15.000 Ptas.

EL BUZO

Javier CANTERO

Spectrum 48 K



Nuestro espíritu aventurero no puede resistir la tentación de embarcarse en una audaz hazaña de la que podemos salir malparados: rescatar un ancestral tesoro oculto en las profundidades del mar.

El tesoro, celosamente guardado durante años por la monumental presencia de un pulpo gigante, se encuentra en una siniestra cueva oceánica sin que, por el momento, haya caído en las manos de aquéllos que lo ambicionaron, todos ellos muertos en el intento.

Nuestro propósito será superar todas las dificultades que entraña la aventura (profundidad del mar, medusas, plantas mortífera, amén de los temibles tentáculos del molusco) y conseguir el, según dicen, fascinante tesoro mediante el hábil manejo de las siguientes teclas:

«Q», arriba.
«A», abajo.
«O», izquierda.
«P», derecha.

NOTAS GRAFICAS

A B C D E F G H I J K L M N O P Q

```
  20 LET ma=0: LET z$="00000"
  30 GO TO 9000
 100 LET d=INT (RND*100)
 110 IF d>(95-sc/100) THEN PRINT
 AT 20,13; INK 5;p$: LET p=1
 200 LET i$=INKEY$
 210 IF i$="a" THEN PRINT AT ky,
kx;" ";AT ky+1,kx;" ": LET a=ATT
R (ky+2,kx). LET c=ATTR (ky+2,kx
+1): GO SUB 1000
 220 IF i$="q" THEN PRINT AT ky,
kx;" ";AT ky+1,kx;" ": LET a=ATT
R (ky-1,kx): LET c=ATTR (ky,kx+1
): GO SUB 1100
 230 IF i$="p" THEN PRINT AT ky,
kx;" ";AT ky+1,kx;" ": LET a=ATT
R (ky,kx+1): LET b=ATTR (ky+1,kx
+1): LET c=ATTR (ky+1,kx+2): GO
SUB 1200
 240 IF i$="o" THEN PRINT AT ky,
kx;" ";AT ky+1,kx," ": LET a=ATT
R (ky,kx-1): LET b=ATTR (ky+1,kx
-1): GO SUB 1300
 250 PRINT INK 7;AT ky,kx;a$,AT
ky+1,kx;b$;: IF t=1 THEN PRINT A
T ky+1,kx+1; INK 6;t$
 280 IF p=1 THEN IF ky=19 AND kx
>12 THEN GO SUB 2100
 290 IF d>(95-sc/100) THEN PRINT
 AT 20,13;"    "; INK 5;"": LET
 p=0
 300 GO TO 100
1000 IF t=1 THEN PRINT AT ky+1,k
x+1;" ": IF c>=11 AND c<=13 THEN
 GO TO 1090
1010 IF a=10 THEN RETURN
1050 IF a=8 THEN LET ky=ky+1: RE
TURN
1090 GO SUB 2100. RETURN
1100 IF t=1 THEN PRINT AT ky+1,k
x+1;" ": IF c>=11 AND c<=13 THEN
 GO TO 1090
1120 IF t=1 THEN IF ky=4 THEN LE
T t=0: GO SUB 2000: RETURN
1130 IF a=8 THEN LET ky=ky-1: RE
TURN
1140 IF a=17 OR a=104 THEN RETUR
N
1190 GO SUB 2100: RETURN
1200 IF t=1 THEN PRINT AT ky+1,k
x+1;" ": IF c>=11 AND c<=13 THEN
 GO TO 1290
1220 IF t=1 AND b=14 THEN LET kx
=kx+1: RETURN
1230 IF a=8 AND b=8 THEN LET kx=
kx+1: RETURN
1270 IF a=10 OR b=10 THEN RETURN
1290 GO SUB 2100: RETURN
1300 IF t=1 THEN PRINT AT ky+1,k
x+1;" "
1310 IF t=0 THEN IF b=14 THEN LE
T t=1: BEEP .02,15: RETURN
1320 IF a=10 OR b=10 OR b=14 OR
a=17 THEN RETURN
1330 IF a=8 AND b=8 THEN LET kx=
kx-1: RETURN
1390 GO SUB 2100: RETURN
2000 BEEP .05,20: BEEP .02,10
2010 LET sc=sc+25. LET s$=STR$ s
c: PRINT AT ky+1,kx+1;" "; PAPER
 5; INK 0; BRIGHT 1;AT 0,21-LEN
s$;s$
2020 RETURN
2100 FOR f=6 TO 0 STEP -1: FOR g
=0 TO 3: BEEP .04,g+f· NEXT g: N
EXT f
2110 IF li=0 THEN GO TO 3000
2120 LET li=li-1: PRINT AT ky,kx
;" ";AT ky+1,kx;" ";AT 0,6; INK
0; PAPER 5; BRIGHT 1;li: LET ky=
4: LET kx=10
2130 IF t=1 THEN PRINT AT ky+1,k
x+1;" ": LET t=0
2140 RETURN
3010 IF sc>ma THEN LET ma=sc: LE
T z$=STR$ ma
3030 PRINT AT 10,5; INVERSE 1; I
NK 6;"QUIERES JUGAR OTRA VEZ?";A
T 12,9,"PULSA (S) O (N)"
3040 LET I$=INKEY$: IF I$="s" TH
EN GO TO 7000
3050 IF i$="n" THEN STOP
3090 GO TO 3040
6000 PAPER 1: INK 6: BORDER 1: C
LS
6010 PRINT TAB 5;"     B U Z
O     "
6020 PRINT : PRINT "   En una cu
eva perdida,se haya un gran teso
ro custodiado por unpulpo gigant
e."
6030 PRINT : PRINT "   Muchos ha
n sido los osados   que lo inten
taron rescatar,pero fracasaron."
6040 PRINT : PRINT "   Ahora tu
tienes la oportuni- dad de resca
tarlo de las profun-didades del
mar."
6050 PRINT : PRINT "   Tus movim
ientos son:"
6060 PRINT : PRINT TAB 11,"Q...A
riba.";TAB 11,"A...Abajo.",TAB 1
1,"O...Izquireda.",TAB 11,"P...D
erecha."
6070 PRINT : PRINT TAB 11;"Buena
 suerte!!"
6080 PRINT #1;" PULSE UNA TECLA
PARA CONTINUAR": PAUSE 0
7000 PAPER 1: INK 0: BORDER 0: B
RIGHT 0: CLS
7005 FOR n=0 TO 3: PRINT AT n,0,
 PAPER 5; BRIGHT 1;"
                    ";AT 3,0;"
 NEXT n
7010 PRINT AT 21,0; INK 2;"[redacted]"
7020 PRINT INK 5;AT 19,18,"",
AT 20,17;""
7030 PRINT INK 2,AT 20,0;"",AT
17,0,"[redacted]";AT 16,0;"[redacted]",AT 15,0,"[redacted]",AT 14,0,"[redacted]",AT 13,
0,"[redacted]",AT 12,0,"[redacted]";
AT 11,0,"[redacted]";AT 10,0,"[redacted]";
AT 9,0,""
7040 PRINT INK 2;AT 20,22,"[redacted]",AT 18,22,"[redacted]"
,AT 17,24,"[redacted]",AT 16,25,
"[redacted]",AT 15,27,"[redacted]"
;AT 14,29;"[redacted]";AT 13,31,"[redacted]
7050 PRINT PAPER 2; INK 1;AT 19,
0;"";AT 18,0;"          ";A
T 19,22;"         "
7060 FOR n=1 TO 4· PRINT INK 4,A
T 5+n,2;"";AT 10+n,10;"";AT 8+
n,7;"";AT 6+n,4;"";AT 13+n,23;
"";AT 11+n,26;"",AT 10+n,28;"
";AT 9+n,30;"": NEXT n
7070 PRINT INK 6;AT 19,2;"";AT
 20,1;"". FOR n=1 TO 10: PRI
NT AT INT (RND*3)+7,INT (RND*20)
+7; INK 3;"": NEXT n
8000 LET ky=4: LET kx=10. LET t=
0: LET li=3: LET sc=0: LET t$="
". LET a$="": LET b$="": LET p
$="": LET p=0
8100 PRINT ; INK 0; PAPER 5; BRI
GHT 1;AT 0,0;"VIDAS:";li;AT 0,9;
"PUNTOS.00000";AT 0,22;"MAX..000
00",AT 0,32-LEN z$,z$
8900 GO TO 100
9000 RESTORE 9020
9010 FOR n=USR "a" TO USR "q"+7:
 READ a: POKE n,a: NEXT n
9020 DATA 60,66,66,66,60,126,219
,219,219,255,60,62,54,54,55,112
9030 DATA 60,66,129,255,73,146,7
3,146,0,0,0,0,0,0,17,238
9040 DATA 0,24,44,6,3,3,1,0,0,0,
1,1,3,131,226,215
9050 DATA 120,204,246,250,250,25
0,254,220,24,36,49,31,0,12,6,3
9060 DATA 60,126,231,129,60,126,
231,129,111,55,255,254,125,27 25
5,254
9070 DATA 252,249,243,181,218,23
6,55,31,96,160,12,26,48,96,224,1
92
9080 DATA 128,216,252,248,250,25
2,255,255,1,49,127,127,63,63,127
,255
9090 DATA 144,84,57,18,212,52,24
,16,90,126,60,60,126,255,255,126
9100 DATA 219,255,60,124,108,108
,236,14
9110 DATA 0,0,0,0,126,215,215,86
9900 GO TO 6000
```


BASKET

Manuel FREIRE

Spectrum 48 K

Premiado con 15.000 Ptas.

Basado en el, por todos conocido, juego de baloncesto, con este programa podrás desarrollar tu destreza mano a mano con el ordenador.

Nada más cargar la cinta aparecerá en pantalla la cancha, fielmente reproducida, en la que dos jugadores, uno con camiseta blanca y el otro con una verde, pondrán a prueba su juego.

Podremos encestar, «taponar» al contrario y desarrollar las jugadas típicas de este deporte, ganando el jugador que antes consiga veinte puntos y finalizando el partido cuando uno de ellos comete cinco personales.

NOTAS GRAFICAS

A B C D E F G H I J K L

```
  10 REM [redacted]
  12 BORDER 0
  15 GO SUB 300
  20 DIM p(2,2)
  60 LET fb=10: LET cb=12. LET f
v=10. LET cv=18
  70 DIM f(4,2): LET f(1,1)=1  I
F RND>.6 THEN LET f(1,1)=0: LET
f (1,2)=1
  80 DIM d$(2,2): LET d$(1)=""
```

```
: LET d$(2)="ÅÅ": IF f(1,1)=1 TH
EN LET d$(1)="♈": LET d$(2)="ÅÅ
"
 100 REM  CIELO
 105 PAPER 0  INK 7: OVER 0: CLS
 : INK 5
 106 BORDER 0: BEEP 0.1,0
 110 PLOT 0,30  DRAW 255,0: DRAW
-50,80: DRAW -155,0  DRAW -50,-
80
 120 PLOT 128,30: DRAW 0,80  CIR
CLE 128,74,12
 130 PLOT 20,60. DRAW 50,10: DRA
W 3,14. DRAW -30,10: CIRCLE 72,7
7,7
 140 PLOT 237,60: DRAW -50,10: D
RAW -3,14: DRAW 30,10: CIRCLE 18
5,77,7
 145 INK 7:
 150 PLOT 12,80. DRAW 20,30: DRA
W 0,30: DRAW -20,-30  DRAW 0,-30
 160 PLOT 19,101: DRAW 7,10: DRA
W 0,10: DRAW -7,-10. DRAW 0,-10
 170 FOR n=1 TO 3: PLOT 21+n,103
+n: DRAW 4,0: NEXT n
 180 PLOT 28,104  DRAW 9,0,3  DR
AW -9,1,2.2
 190 FOR n=3 TO 7 STEP 2: PLOT 3
7,103  DRAW -n,-10: PLOT 28,103.
 DRAW n,-10: NEXT n
 195 INK 4
 200 PLOT 245,80: DRAW -20,30. D
RAW 0,30: DRAW 20,-30: DRAW 0,-3
0
 210 PLOT 238,101: DRAW -7,10  D
RAW 0,10  DRAW 7,-10  DRAW 0,-10
 220 FOR n=1 TO 3  PLOT 236-n,10
3+n: DRAW -4,0: NEXT n
 230 PLOT 229,104: DRAW -9,0,-3.
 DRAW 9,1,-2.2
 240 FOR n=3 TO 7 STEP 2. PLOT 2
20,103. DRAW n,-10  PLOT 229,103
: DRAW -n,-10  NEXT n
 250 PRINT AT 1,5, INK 7,"PUNTO
S",AT 1,20, INK 4,"PUNTOS"
 260 PRINT AT 1,14, INK 7,"00",A
T 1,17; INK 4,"00"
 270 PRINT AT 3,6; INK 7,"Pers",
AT 3,23, INK 4,"Pers"
 280 PRINT AT 3,12; INK 7;"0";AT
 3,20, INK 4;"0"
 282 OVER 1: INK 5. PLOT 30,170:
 DRAW 200,0: DRAW 0,-28: DRAW -2
00,0: DRAW 0,28
 284 PLOT 29,171: DRAW 200,0: PL
OT 29,171: DRAW 0,-28
 285 PLOT 28,172. DRAW 200,0. PL
OT 28,172. DRAW 0,-28
 290 INK 8: OVER 1. GO TO 500
 300 REM  Def. Graficos
 310 FOR n=1 TO 12  READ a$  FOR
m=0 TO 7. READ a  POKE USR a$+m
,a: NEXT m. NEXT n
 320 DATA "a",2,2,114,114,114,36
,40,48
 330 DATA "b",48,48,48,120,72,72
,72,108
 340 DATA "c",0,0,112,112,115,36
,59,51
 350 DATA "d",48,48,48,40,68,130
,130,195
 360 DATA "e",0,0,112,112,112,39
,56,48
 370 DATA "f",48,48,48,120,72,72
,75,111
 380 DATA "g",64,64,78,78,78,36,
20,12
 390 DATA "h",12,12,12,30,18,18,
18,54
 400 DATA "i",0,0,14,14,206,36,2
20,204
 410 DATA "j",12,12,12,26,34,65,
65,195
 420 DATA "k",0,0,14,14,14,228,2
8,12
 430 DATA "l",12,12,12,30,18,18,
210,246
 450 RETURN
 500 REM  BUCLE PRINCIPAL
 502 PRINT INK 7,AT fb,cb,d$(1,1
),AT fb+1,cb,d$(2,1)
 504 PRINT INK 4;AT fv,cv,d$(1,2
),AT fv+1,cv,d$(2,2)
 505 REM   ** Movimiento **
 507 LET fb2=fb. LET cb2=cb: LET
 fv2=fv: LET cv2=cv
 510 LET fb=fb+(IN 65022=253)-(I
N 64510=253)
 515 IF fb>16 OR fb<7 THEN LET f
b=fb2
 520 LET cb=cb+(IN 64510=247)-(I
N 64510=251)
 530 IF fb=fv AND cb=cv THEN GO
TO 3100
 540 LET fv=fv+(IN 49150=251)-(I
N 57342=253)
 545 IF fv>16 OR fv<7 THEN LET f
v=fv2
 550 LET cv=cv+(IN 32766=251)-(I
N 32766=247)
 560 IF fb=fv AND cb=cv THEN GO
TO 3200
 570 GO SUB 700
 690 GO TO 505
 700 REM  JUGADORES
 710 LET f(4,1)=1   IF fb2=fb AND
 cb2=cb THEN LET f(4,1)=0
 720 LET f(4,2)=1. IF fv2=fv AND
 cv2=cv THEN LET f(4,2)=0
 750 PRINT INK 7,AT fb2,cb2,d$(1
,1);AT fb2+1,cb2;d$(2,1)
 752 IF f(2,1)<>0 OR IN 65278=25
1 THEN GO TO 1000
 755 IF f(1,1)=0 THEN GO TO 780
 760 IF d$(1,1)="♈" THEN LET d$(
1,1)="♈": LET d$(2,1)="Å": BEEP
.001,20: GO TO 800
 770 LET d$(1,1)="♈": GO TO 790
 780 LET d$(1,1)="♈"
 790 LET d$(2,1)="Å": IF f(4,1)=
1 THEN LET d$(2,1)="Å"
 800 PRINT INK 7,AT fb,cb,d$(1,1
);AT fb+1,cb;d$(2,1)
 850 PRINT INK 4;AT fv2,cv2;d$(1
,2),AT fv2+1,cv2,d$(2,2)
 852 IF f(2,2)<>0 OR IN 32766=25
3 THEN GO TO 2000
 855 IF f(1,2)=0 THEN GO TO 880
 860 IF d$(1,2)="♈" THEN LET d$(
1,2)="♈": LET d$(2,2)="Å": BEEP
.001,20: GO TO 900
 870 LET d$(1,2)="♈": GO TO 890
 880 LET d$(1,2)="♈"
 890 LET d$(2,2)="Å": IF f(4,2)=
1 THEN LET d$(2,2)="Å"
 900 PRINT INK 4;AT fv,cv;d$(1,2
);AT fv+1,cv;d$(2,2)
 950 RETURN
1000 REM  Salto Disparo B
1010 IF f(1,1)=1 THEN GO TO 1100
1020 IF f(2,1)=1 THEN LET f(2,1)
=f(2,1)-1: LET fb=fb+1: GO TO 80
0
1030 IF f(2,1)>1 THEN LET f(2,1)
=f(2,1)-1: GO TO 800
1040 REM   *** Salto ***
1045 LET f(2,1)=3
1050 LET fb=fb-1
1060 LET d$(1,1)="♈". LET d$(2,1
)="Å"
1070 GO TO 800
1100 REM  *** Disparo ***
1105 IF cb<15 THEN GO TO 755
1110 LET a=8+(cb*8)  LET b=175-(
fb*8)
1115 LET e=(cv*8): LET f=175-(fv
*8)
1120 LET c=230: LET d=150
1125 LET d$(1,1)="♈". LET d$(2,1
)="Å"  PRINT INK 7,AT fb,cb;d$(1
,1);AT fb+1,cb;d$(2,1)
1130 LET x=a: LET y=b
1135 PLOT x,y: PLOT x+1,y: PLOT
x,y+1: PLOT x+1,y+1
1140 PLOT x,y: PLOT x+1,y: PLOT
x,y+1: PLOT x+1,y+1
1150 LET x=x+2: LET y=y+2*((d-b)
/(c-a))
1160 PLOT x,y: PLOT x+1,y: PLOT
x,y+1: PLOT x+1,y+1
1170 IF ABS (x-e)<3 AND ABS (y-f
)<3 THEN GO TO 3600
1190 IF IN 65278=251 THEN GO TO
1140
1200 PLOT x,y. PLOT x+1,y. PLOT
x,y+1  PLOT x+1,y+1
1210 LET x=x+2  LET y=y-2*((d-b)
/(c-a))
1220 PLOT x,y: PLOT x+1,y: PLOT
x,y+1  PLOT x+1,y+1
1230 IF y<95 THEN PLOT x,y: PLOT
 x+1,y. PLOT x,y+1. PLOT x+1,y+1
: GO TO 3600
1240 IF x=232 THEN GO TO 1300
1250 IF ABS (225-x)<3 AND ABS (1
04-y)<3 THEN GO TO 3300
1290 GO TO 1200
1300 PLOT x,y  PLOT x+1,y: PLOT
x,y+1  PLOT x+1,y+1
1310 LET x=x-2: LET y=y-2*((d-b)
/(c-a))
1320 PLOT x,y  PLOT x+1,y  PLOT
x,y+1. PLOT x+1,y+1
1330 IF y<95 THEN PLOT x,y: PLOT
 x+1,y: PLOT x,y+1: PLOT x+1,y+1
: GO TO 3600
1340 IF ABS (225-x)<4 AND ABS (1
04-y)<4 THEN GO TO 3300
1350 GO TO 1300
2000 REM  Salto Disparo V
2010 IF f(1,2)=1 THEN GO TO 2100
2020 IF f(2,2)=1 THEN LET f(2,2)
=f(2,2)-1  LET fv=fv+1  GO TO 90
0
2030 IF f(2,2)>1 THEN LET f(2,2)
=f(2,2)-1: GO TO 900
2040 REM   *** Salto ***
2045 LET f(2,2)=3
2050 LET fv=fv-1
2060 LET d$(1,2)="♈": LET d$(2,2
)="Å"
2070 GO TO 900
2100 REM  *** Disparo ***
2105 IF cv>15 THEN GO TO 855
2110 LET a=(cv*8): LET b=175-(fv
*8)
2115 LET e=8+(cb*8)  LET f=175-(
fb*8)
2120 LET c=22: LET d=150
2125 LET d$(1,2)="♈"  LET d$(2,2
)="Å": PRINT INK 4,AT fv,cv;d$(1
,2),AT fv+1,cv,d$(2,2)
2130 LET x=a  LET y=b
2135 PLOT x,y  PLOT x+1,y  PLOT
x,y+1  PLOT x+1,y+1
2140 PLOT x,y  PLOT x+1,y  PLOT
x,y+1  PLOT x+1,y+1
2150 LET x=x-2: LET y=y+2*(ABS (
d-b)/ABS (c-a))
2160 PLOT x,y: PLOT x+1,y: PLOT
x,y+1  PLOT x+1,y+1
2170 IF ABS (x-e)<3 AND ABS (y-f
)<3 THEN GO TO 3500
2190 IF IN 32766=253 THEN GO TO
2140
2200 PLOT x,y  PLOT x+1,y  PLOT
x,y+1: PLOT x+1,y+1
2210 LET x=x-2. LET y=y-2*(ABS (
d-b)/ABS (c-a))
2220 PLOT x,y: PLOT x+1,y: PLOT
x,y+1: PLOT x+1,y+1
2230 IF y<95 THEN PLOT x,y: PLOT
 x+1,y  PLOT x,y+1  PLOT x+1,y+1
  GO TO 3500
2240 IF x=26 THEN GO TO 2300
2250 IF ABS (32-x)<3 AND ABS (10
4-y)<3 THEN GO TO 3400
2290 GO TO 2200
2300 PLOT x,y. PLOT x+1,y: PLOT
x,y+1: PLOT x+1,y+1
2310 LET x=x+2: LET y=y-2*(ABS (
d-b)/ABS (c-a))
2320 PLOT x,y: PLOT x+1,y: PLOT
x,y+1  PLOT x+1,y+1
2330 IF y<95 THEN PLOT x,y  PLOT
 x+1,y: PLOT x,y+1: PLOT x+1,y+1
: GO TO 3500
2340 IF ABS (32-x)<4 AND ABS (10
4-y)<4 THEN GO TO 3400
2350 GO TO 2300
3000 REM  PERSONALES
3100 REM Blanco
3105 OVER 0
3110 PRINT AT 5,10; INK 7; FLASH
 1;" PERSONAL "
3120 LET p(2,1)=p(2,1)+1
3130 PRINT AT 3,12, INVERSE 1; I
NK 7, FLASH 1,p(2,1)
3140 FOR n=1 TO 150  NEXT n
3150 PRINT AT 5,10, INK 7, FLASH
 0,"          "
3160 PRINT AT 3,12; INK 7; FLASH
 0,p(2,1)
3170 IF p(2,1)=5 THEN STOP
3172 LET fb=fb2  LET fv=fv2  LET
 cb=cb2  LET cv=cv2
3175 OVER 1
3180 GO TO 3600
3200 REM Verde
3205 OVER 0
3210 PRINT AT 5,10; INK 4; FLASH
 1," PERSONAL "
3220 LET p(2,2)=p(2,2)+1
3230 PRINT AT 3,20; INVERSE 1; I
NK 7; FLASH 1;p(2,2)
3240 FOR n=1 TO 150: NEXT n
3250 PRINT AT 5,10; INK 7; FLASH
 0;"          "
3260 PRINT AT 3,20; INK 7; FLASH
 0;p(2,2)
3270 IF p(2,2)=5 THEN STOP
3272 LET fb=fb2: LET fv=fv2: LET
 cb=cb2: LET cv=cv2
3275 OVER 1
3280 GO TO 3500
3300 REM  CANASTAS
3305 OVER 0
3310 REM Blanco
3320 PRINT AT 5,10, INK 7, FLASH
 1," CANASTA "
3330 LET p(1,1)=p(1,1)+2  LET a$
=STR$ p(1,1)  IF p(1,1)<10 THEN
LET a$="0"+a$
3340 PRINT AT 1,14, INK 7, FLASH
 1, INVERSE 1,a$
3350 FOR n=1 TO 150: NEXT n
3360 PRINT AT 5,10, INK 7, FLASH
 0,"          "
3370 PRINT AT 1,14; INK 7, FLASH
 0,a$
3380 IF p(1,1)=20 THEN STOP
3385 OVER 1: PLOT x,y: PLOT x+1,
y  PLOT x,y+1  PLOT x+1,y+1
3390 GO TO 3600
3410 REM Verde
3415 OVER 0
3420 PRINT AT 5,10; INK 4; FLASH
 1," CANASTA "
3430 LET p(1,2)=p(1,2)+2: LET a$
=STR$ p(1,2)  IF p(1,2)<10 THEN
LET a$="0"+a$
3440 PRINT AT 1,17, INK 4, FLASH
 1, INVERSE 1,a$
3450 FOR n=1 TO 150  NEXT n
3460 PRINT AT 5,10, INK 4; FLASH
 0,"          "
3470 PRINT AT 1,17; INK 4; FLASH
 0,a$
3480 IF p(1,2)=20 THEN STOP
3485 OVER 1: PLOT x,y: PLOT x+1,
y: PLOT x,y+1: PLOT x+1,y+1
3490 GO TO 3500
3500 REM  SAQUE
3510 REM Blanco
3520 LET f(1,1)=1: LET f(1,2)=0
3525 PRINT INK 7;AT fb,cb;d$(1,1
),AT fb+1,cb,d$(2,1)
3527 PRINT INK 4,AT fv,cv,d$(1,2
);AT fv+1,cv;d$(2,2)
3530 LET fb=10  LET cb=5  LET fv
=10: LET cv=14
3540 GO TO 500
3610 REM Verde
3620 LET f(1,1)=0. LET f(1,2)=1
3625 PRINT INK 7,AT fb,cb;d$(1,1
),AT fb+1,cb,d$(2,1)
3627 PRINT INK 4,AT fv,cv;d$(1,2
);AT fv+1,cv;d$(2,2)
3630 LET fb=10: LET cb=16: LET f
v=10: LET cv=27
3640 GO TO 500
```

COPYUPI, UN COPIADOR MUY ESPECIAL (y II)

Lorenzo CEBEIRA

En ésta segunda y última parte vamos a mostraros el manejo y uso de las rutinas para su adecuada utilización, así como el desensamble de las mismas para una mejor comprensión.

Y vamos ya con las rutinas. En la tabla adjunta se muestran las direcciones de memoria donde se ubican las distintas subrutinas y los datos más relevantes.

Estas están situadas en el Buffer de la impresora, lo que nos deja libre toda la memoria de programa como zona de trabajo para COPYUPI. Así, por ejemplo, si queremos cargar un programa normal haremos PRINT USR 23373 y si queremos grabarlo haremos PRINT USR 2335Ø. Pero antes de llamar a las rutinas hay que dejar sitio libre en memoria para cargar los programas. COPYUPI utiliza como zona de trabajo toda la memoria por encima de RAMTOP, por lo que si queremos cargar el programa en, digamos, 3ØØØØ habrá previamente que hacer CLEAR 29999. Otra consideración a tener en cuenta es que las rutinas de carga informan del resultado con tres posibilidades. Ø significa correcto, 1 que el programa no cabe en la memoria que hemos reservado y 2 que se ha producido un error. La única excepción a la regla es la rutina de carga sin cabecera (señalada como LOAD ESPECIAL) que puede devolver un Ø sin haber cargado nada. En este caso hay que comprobar que la longitud del programa que se ha cargado sea diferente de Ø. Esta longitud la devuelve COPYUPI en los ya conocidos bytes 12 y 13 de la cabecera que hemos visto antes. Y vamos ya a analizar cada rutina por separado.

LOAD NORMAL

La rutina de LOAD NORMAL sirve para cargar un programa normal con cabecera. Esta cabecera se almacenará en la dirección 23522 y siguientes, donde podremos analizarla tranquilamente. El resto del programa se cargará justo por encima de RAMTOP como ya hemos visto. Para utilizarla haremos PRINT USR 23373 y, como ya sabemos, si se imprime Ø en la pantalla significará que se ha cargado el programa correctamente, un 1 que el programa es demasiado largo para la memoria libre y un 2 que ha habido un error de carga.

LOAD ESPECIAL

La rutina de LOAD ESPECIAL carga los programas sin cabecera, a condición de que el flag esté debidamente ajustado (más adelante veremos cómo dar al flag el valor adecuado). Esta rutina devuelve la longitud cargada en las direcciones correspondientes de la cabecera que se indican al final del listado. A diferencia de la anterior, esta rutina no indica si el programa ha resultado demasiado largo para la memoria disponible por lo que habrá que calcularlo manualmente. Como de costumbre, nos delvolverá un Ø para indicar operación correcta, con la salvedad que señalamos más arriba, y un 2 si ha habido error de carga.

SAVE NORMAL

La rutina de SAVE NORMAL sirve para grabar un programa con su correspondiente cabecera. Una de las ventajas de COPYUPI es que podemos emplear esta función para grabar cualquier cosa que tengamos en la memoria. La aplicación más corriente es cargar un programa sin cabecera, ponerle la cabecera que queramos y grabarlo como un programa normal. Otra posibilidad es, cargar un programa normal, eliminar la ejecución automática modificando la cabecera y grabarlo con esta rutina. También podemos modificar RAMTOP para grabar sólo parte del programa cargado o para fundir dos programas en uno.

VERIFICAR NORMAL

La rutina de VERIFICAR NORMAL se utiliza para comprobar que la copia del programa ha sido correcta. Devuelve un Ø en este caso y un 2 si ha habido algún error. En cuanto a la rutina de verificar especial realiza la misma función con los programas sin cabecera.

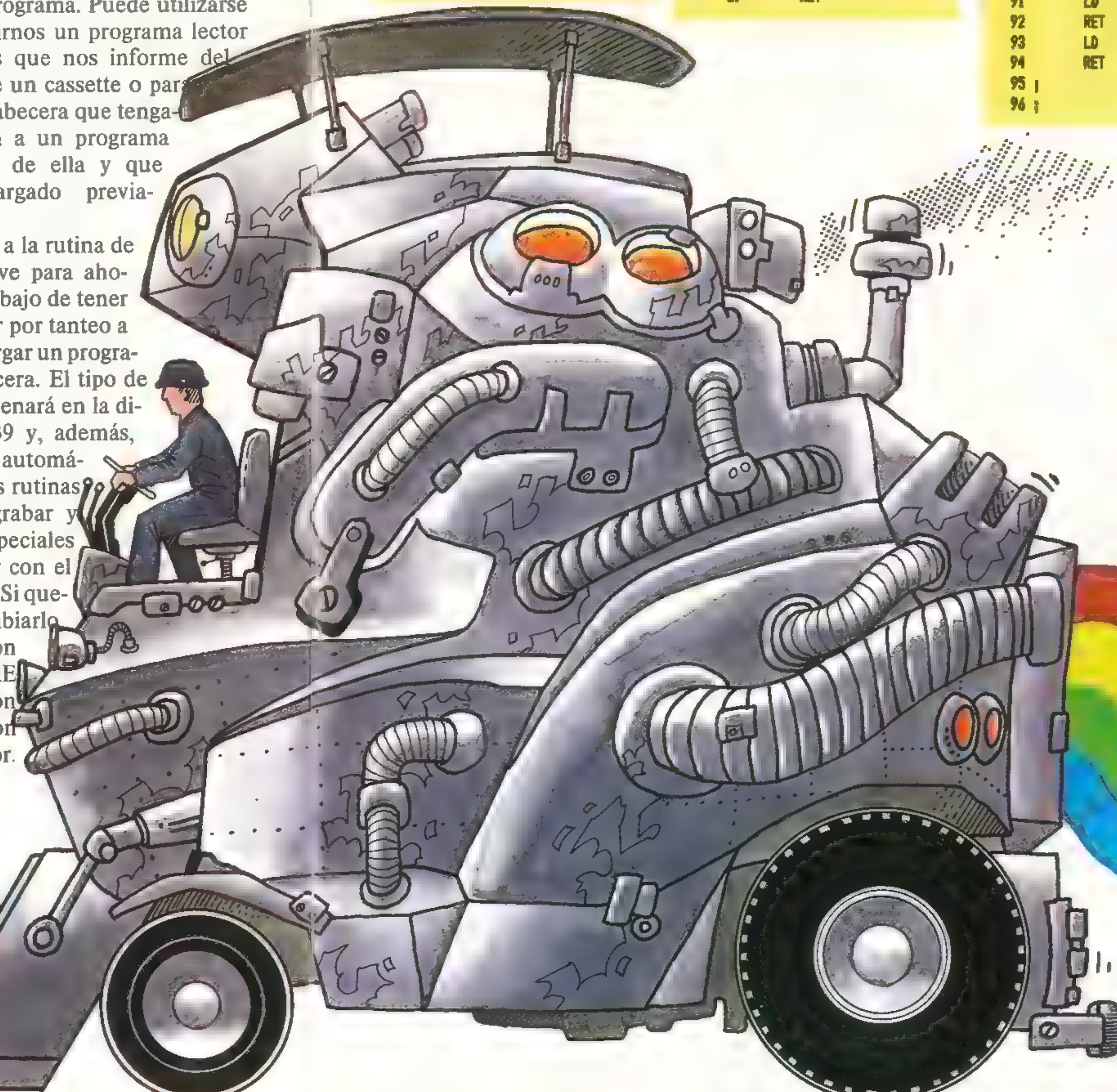
Lector de cabeceras y flags

La rutina de leer cabeceras es una función que se limita a cargar la cabecera de un programa. Puede utilizarse para construirnos un programa lector de cabeceras que nos informe del contenido de un cassette o para añadir una cabecera que tengamos grabada a un programa que carezca de ella y que hayamos cargado previamente.

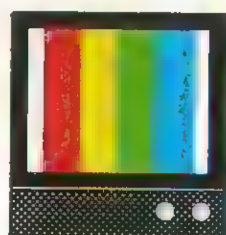
En cuanto a la rutina de leer flag, sirve para ahorrarnos el trabajo de tener que proceder por tanteo a la hora de cargar un programa sin cabecera. El tipo de flag se almacenará en la dirección 23539 y, además, se ajustarán automáticamente las rutinas de cargar, grabar y verificar especiales para trabajar con el nuevo valor. Si queremos cambiarlo bastará con hacer POKE en la dirección indicada con el nuevo valor.

DESENSAMBLE DE LAS RUTINAS

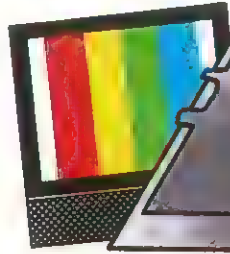
```
#D+
#C-
;
;
        ORG  23296
;
;
TIPO    CALL ADJUST
F3      LD   A,#00
        LD   DE,0
        LD   IX,HEADER
        CALL #0556
        LD   A,L
        LD   (FLAG),A
        CALL ADJUST
        LD   B,#00
        LD   C,A
        RET
;
;
VERIFY  CALL PREPAC
        CALL #0556
        LD   BC,#0002
        RET  NC
        LD   A,#FF
        LD   (FLAG),A
VERX    CALL PREPAR
        CALL #0556
        LD   BC,#0002
        RET  NC
        LD   C,#00
        RET
;
;
SAVE    CALL PREPAC
        CALL #04C2
        LD   B,50
LOOP    HALT
        DJNZ LOOP
        LD   A,#FF
        LD   (FLAG),A
DSAVE   CALL PREPAR
        CALL #04C2
        RET
;
;
LOAD    CALL CARCAB
        CALL COMCAB
        LD   A,0
        CP   C
        RET  C
        CALL PREPAR
        LD   A,#FF
        SCF
        CALL #0556
        LD   BC,#0002
        RET  NC
        LD   C,#00
        RET
;
;
PREPAC  LD   A,#00
        LD   DE,#11
        LD   IX,HEADER
        RET
;
;
CARCAB  CALL PREPAC
        SCF
        CALL #0556
        JR   NC,CARCAB
        LD   A,(HEADER)
        CP   #04
        JR   NC,CARCAB
        RET
;
;
ADJUST  LD   A,(FLAG)
        LD   (F1+1),A
        LD   (F2+1),A
        LD   (F3+1),A
        RET
;
;
COMCAB  AND  A
        LD   HL,(23732)
        LD   BC,(23730)
        SBC  HL,BC
        LD   BC,(HEADER+11)
        SBC  HL,BC
        LD   BC,#0001
        RET  C
        LD   C,#00
        RET
;
;
PREPAR  CALL ADJUST
F1      LD   A,#FF
        LD   DE,(HEADER+11)
        LD   IX,(23730)
        INC  IX
        RET
;
;
DLOAD   CALL PREPAR
        LD   HL,(23732)
        PUSH IX
        POP  BC
        AND  A
        SBC  HL,BC
        PUSH HL
        POP  DE
F2      LD   A,#FF
        SCF
        CALL #0556
        LD   BC,#0002
        LD   A,H
        CP   #00
        RET  NZ
        PUSH IX
        POP  HL
        DEC  HL
        LD   BC,(23730)
        SBC  HL,BC
        LD   (HEADER+11),HL
        LD   BC,#0000
        RET
;
;
HEADER  DEFS 17
;
;
FLAG    DEFS 1
```











F. L.FRONTÁN

Ampliación de memoria

¿Existe la posibilidad de ampliar el Spectrum 48 K RAM a más memoria RAM o está condenado a quedarse con tan poca capacidad de memoria?
¿Se puede ampliar la ROM del Spectrum exteriormente como el QL?
¿Por qué los G.D.U. se direccionan en la memoria del ordenador con las letras de la "A" a la "U" y no las restantes del abecedario? ¿Los G.D.U. se sitúan en ROM o en RAM? ¿Si se almacenan en la ROM, se pierden al apagar el ordenador?
¿Cuándo se usa "CODE" para grabar un programa? ¿Tiene alguna utilidad REM aparte de memorizar o recordar lo que se está haciendo? ¿El Spectrum tiene Compilador o Intérprete? ¿Cuántos programas se pueden grabar en una misma cinta?
¿Hay alguna casa distribuidora de Spectrum que haga la ampliación de memoria?

Agustín LOPEZ - Parla

□ El microprocesador Z-80 sólo permite direccionar 64 K de memoria tanto RAM como ROM, por tanto, cualquier ampliación a más de 64 K exigiría paginar la memoria, le remito al artículo aparecido en el número 5 de esta revista.

Los G.D.U. ocupan las 168 últimas posiciones de la memoria RAM, como cada uno ocupa 8 posiciones sólo hay sitio para 21 G.D.U. es decir, de la "A" a la "U" y se pierden al apagar el ordenador.

Se usa "CODE" para grabar un programa cuando éste está escrito en Código Máquina, pero "CODE" también sirve para grabar una serie de bytes, por ejemplo los G.D.U. o la pantalla. La sentencia "REM" sólo sirve para incluir comentarios en un programa. El Spectrum utiliza un intérprete, por lo que trabaja siempre sobre el código fuente. El número de programas que cabe en una cinta depende de la longitud de los mismos, a título indicativo, en una C-60 caben más de 30 programas de 16 K.

La misma firma que importa el Spectrum se encarga de ampliarle la memoria.

Contabilidad en el Spectrum

Les escribo esta carta para preguntarles si en el Spectrum de 16 o 48 K se pueden llevar la contabilidad y ficheros de un pequeño negocio, y si se venden cintas con programas comerciales.

Rafael MARQUEZ - Córdoba

□ Efectivamente, el Spectrum permite llevar contabilidades y ficheros. Para ello deberá usar la versión de 48 K, así como algún dispositivo de almacenamiento masivo de datos (Disco o Microdrive). Existe comercialmente software al efecto, adaptado a Microdrive (Bases de datos y Plan General de Contabilidad).

Impresoras

Estoy pensando en comprarme una impresora y tenía elegidas la SEIKOSHA GP-550 o bien la STAR GEMINI 10X. Deseo que me aclaren, si está en su mano, cuál es la forma de conectar estas impresoras al Spectrum para obtener el máximo provecho de ellas.
¿Cómo puedo averiguar las direcciones de comienzo y fin de un programa?, lo necesito para un desensamblador que me pide estos datos.

José I. HENCHE - Huesca

□ Las impresoras que nos indica utilizan la norma CENTRONICS, por lo que necesitará un interface para impresora de este tipo. Para obtener el máximo rendimiento, deberá seguir las instrucciones del interface que elija. Le recomendamos el de INDESCOMP por ser el de manejo más sencillo, con la ventaja adicional de ser compatible con todos los procesadores de textos existentes para el Spectrum. Junto con el interface, deberá adquirir un cable CENTRONICS para conectarlo con la impresora.

Para averiguar la dirección de comienzo y la longitud de un programa, puede utilizar el programa «Listador» publicado en el número 2, página 10 de nuestra revista. Caso de que el programa en cuestión no tenga cabecera, deberá desensamblar primero la rutina que lo carga, para averiguar el valor que introduce en el registro «IX» (inicio) y en el «DE» (longitud).

Teclados profesionales

Quisiera preguntar dos cuestiones sobre el ZX Spectrum Plus.
1. ¿Se le puede acoplar un teclado distinto aunque no fuera profesional, por ejemplo el del 48 K?
2. ¿Se puede meter un programa escrito para 16 K?

Luis MORENO - Málaga

□ Al Spectrum Plus se le pude acoplar cualquier teclado que se le pueda acoplar al Spectrum, ya que la placa de circuito impreso de ambos ordenadores es la misma y sólo se diferencian en la carcasa.

Por otro lado, le puede introducir programas escritos para 16 K, el único problema podría surgir con la dirección de los UDG si se pone de forma absoluta en lugar de utilizar la función USR "a".

La ZX-Printer

Deseo comprar una impresora ZX-Printer usada, pues tengo entendido que ya no existe en el mercado, y me gustaría que me informasen sobre las prestaciones de que dispone. Así mismo les ruego me informen sobre la existencia o no del papel a utilizar y el servicio técnico que existe.

Francisco ABELLAN - Murcia

□ Efectivamente, la ZX-Printer ha dejado de fabricarse, por lo que le recomendamos que opte por la SEIKOSHA GP-50S o la ALPHACOM-32, ambas de precio y prestaciones similares a la ZX-Printer. La GP-50S tiene la ventaja adicional de utilizar papel normal (no térmico).

Son impresoras matriciales de 32 columnas con una sola aguja de impresión, sin buffer (utilizan el del ordenador) y más bien lentas; aunque eso sí, son muy baratas.

La ZX-NET

He leído que pueden conectar más de un Spectrum (hasta 64 unidades), para potenciar sus posibilidades.
¿Me podrían explicar de qué forma puedo conectar dos Spectrum y si realmente se obtiene la teórica cifra de 96 K de memoria?

Ricardo CASTRO - Sevilla

□ Es posible conectar varios Spectrum (hasta 64) a través de lo que Sinclair ha denominado la «ZX-NET», para ello necesitará el Interface 1 de Sinclair.

La conexión de dos Spectrum en una red no aumenta la memoria de ninguno de ellos. Cada uno sigue teniendo 48 K de memoria RAM, pero se pueden intercambiar datos y programas entre ellos, con lo que pueden compartir periféricos (impresora, Microdrive, etc.).

Más de 21 UDG

¿Hay alguna manera de hacer más de 21 gráficos definidos por el usuario?

Luis SOTILLOS - Madrid

□ Como ya hemos comentado otras veces, existen varios trucos que permiten trabajar con más de 21 UDGs. Todos ellos requieren un buen conocimiento del ordenador, e implican la alteración de ciertas variables del sistema (CHARS o UDG).

Las amplias explicaciones que requieren se salen del reducido espacio disponible en esta sección, pero son temas que hemos tratado y seguiremos tratando en diversos artículos de la revista.

Funciones definibles

Desearía que me explicaran la utilización de las teclas «DEF FN» y «FN».

Fernando GARCIA - Cádiz

□ Se trata de las teclas que controlan las funciones definibles del Spectrum. El comando DEF FN sirve para definir una función que el Spectrum usará como una de las propias que ya tiene definidad. FN sirve para llamar a esa función. Como ejemplo, supongamos que queremos hayar el cuadrado de un número multiplicado por 2, haríamos:

```
1Ø DEF FN a(x)=2*x↑2
2Ø INPUT n
3Ø LET y=FN a(n)
4Ø PRINT y: GO TO 2Ø
```

ZX-Microdrive

Tengo interés en comprarme un ZX-Microdrive, ¿cuáles son sus ventajas y cuáles sus inconvenientes?

José V. GISPERT - Madrid

□ El microdrive es un dispositivo de almacenamiento masivo de acceso secuencial, si bien al estar formateado, da al usuario la impresión de un acceso aleatorio.

La ventaja principal de trabajar con un dispositivo de almacenamiento de este tipo, es la rapidez en la grabación y recuperación de datos y programas.

Refiriéndonos concretamente al Microdrive, su mayor ventaja es el precio. En cuanto a los inconvenientes, se podrían citar la relativa lentitud de acceso, la escasa fiabilidad (si lo comparamos con un disco) y la difícil sintaxis de sus instrucciones al no disponer de un sistema operativo propio.

Problemas matemáticos

Intento hacer un programa que me haye todos los números de tres cifras tales que la suma de los cubos del valor absoluto de sus cifras, coincida con el número, el programa es el siguiente:

```
2Ø FOR a=1 TO 9
3Ø FOR b=Ø TO 9
4Ø FOR c=Ø TO 9
5Ø LET n=1ØØ*a+1Ø*b+c
6Ø LET x=a↑3+b↑3+c↑3
7Ø IF n<>x THEN GO TO 9Ø
8Ø PRINT n
9Ø NEXT c: NEXT b: NEXT a
```

*Con este programa no sale nada, ¿por qué? Si se cambia la línea 6Ø POR: 6Ø LET a=a*a*a+b*b*b+c*c*c entonces el programa se ejecuta y salen los resultados: 153, 37Ø, 371 y 4Ø7.*

A. PEREZ - Madrid

□ El fallo se debe a que la exponenciación funciona por logaritmos, con lo que se produce un error de redondeo; error que tiene consecuencias nefastas cuando se intenta la comparación. Otra forma de hacerlo funcionar es incluir la línea: 65 LET x=VAL STR$ x con lo que queda anulado el error de redondeo.

Programando en C/M

¿Qué debo hacer para programar mi Spectrum en C/M, es necesario algún tipo de software para ello?

Pedro CHACOPINO - Baleares

□ Los programas en código máquina se escriben primero en lenguaje Assembler y luego, se ensamblan en código máquina. El proceso de ensamblado se puede hacer en mano si el programa no es demasiado largo, o utilizar un «Ensamblador», que es un programa que traduce el código en Assembler (código fuente) a código máquina (código objeto).

DE OCASION

● VENDO ordenador ZX Spectrum 48K por 33.000 ptas., con el libro «Diseño y programación de juegos». Alrededor de 50K de uso. Vendo también una máquina de escribir portatil casi nueva (6 meses de uso) por 9.000 ptas. Todo junto por 40.000 ptas. (Para comprar un plus). Interesados llamar al Tel.: (91)2661909, preguntar por José Manuel.

● COMPRO sintetizador de voz para el Spectrum por menos de 3.000 ptas. Los interesados escriban a: Paco Gonzalo Vilar. Avda. de los Castros, 38, 9.° Santander, o bien, llamar al Tel.: 276247. Cantabria.

● CLUB del Spectrum, hace ampliación de socios, revista mensual, cambio de ideas, consultorio, concursos, etc. Todo gratis. Escribe a: Data Compute. Cuesta de S. Vicente, 18. 28008 Madrid.

● VENDO impresora Seikosha 250 X, más interface Centronics para Spectrum, cable por 40.000 ptas. y regalo cinta Context (procesador de textos). Interesados llamar al Tel.: 262083 de Lérida.

● VENDO ZX Spectrum de 16K, sin estrenar todavía. Incluyo además la cinta Horizontes, alimentador, manual de Basic y el de instrucciones en castellano, cables y conexiones por 25.000 ptas. Carlos González. Tel.: 7110820 de Madrid.

● VENDO ZX Spectrum de 16K, con todos los cables y fuente de alimentación. Regalo cable de conexión para TV de antena de 4 salidas, y revistas. Todo por 30.000 ptas. Llamar al Tel.: (958)294834. Preguntar por Luis Alberto Sanz.

● VENDO Spectrum 48K, completo, en buen estado, más revistas, todo por 29.000 ptas. También vendo ordenador de ajedrez «Chess Challenger», 8 niveles, cables y transformador por 25.000 ptas. Contactar con Ramón A. Valero Crespo. Avda. Monforte de Lemos, 75, 4.° C. 28029 Madrid. Tel.: 7385673.

● VENDO ordenador 48K Spectrum, comprado en nov. 84, con todos los accesorios (transformador, cables, manual, etc.). El manual se dará en las dos versiones (el original en inglés y el libro traducido al español). Precio: 33.000 ptas. Preguntar por Victor en el Tel.: 2011627 de Madrid.

● VENDO sintetizador de voz y Spectrum 48K y dos libros, por el precio de 35.000 ptas., con todos los accesorios y manuales. Llamar al Tel.: (972)369246. Preguntar por Pedro Morales.

● DESEARIA recibir manual de código máquina para el Spectrum, manual de Cobol, de Forth y del Msx (pueden ser fotocopias). Interesados llamar al Tel.: 6716118 de Coslada (Madrid).

● VENDO ZX Spectrum Plus, adquirido en ene. 85, con accesorios. Precio: 42.000 ptas. Santiago Soto-Largo. Narváez, 86. 28009 Madrid. Tel.: 2743468.

● VENDO Ordenador ZX Spectrum 48K, muy barato, completo (manual en castellano, fuente de alimentación, cables...) con TV 12", cassette especial, revistas, Vendo todo completo, nuevo, casi sin usar, a personas que vivan en Zaragoza. Interesados escribir a: Arturo Ramón. Gral. Sueiro, 39, 6.° C. 50008 Zaragoza.

● VENDO «Mesa Sund» para Spectrum, por estrenar, precio de 3.000 ptas. Interesados llamar al Tel.: (952)219019 de Málaga (llamar tardes).

● VENDO interface de Joystick programables Indescomp, en perfecto estado, con instrucciones incluidas. Se puede controlar con Joystick cualquier juego. Interesados llamar al Tel.: 6680672. Molins de Rei. Barcelona. Preguntar por Jordi.

● VENDO ZX Spectrum 48K, con cables, alimentador eléctrico, cinta Horizontes, más dos libros para el Spectrum: «ZX Spectrum: ¿Qué es, cómo se usa, para qué sirve?» y «Aplicaciones para la casa y los pequeños negocios». Todo ello por 29.000 ptas. Interesados escribir a: Eduardo Ruiz Hurtado. Avda. José Antonio, 7, 2.° D. 24001 León.

● VENDO ordenador ajedrez miniatura, magnético, 4 niveles, diseño y fabricación ingleses, 10.000 ptas. Calculadora programables Texas, Instruments TI-59, completa (cargador, baterias, tarjetas, etc.) por 25.000 ptas. o bien lo Cambio por radio control 6 canales. Tel.: (96)3499696. Juan Antonio.

● CAMBIO proyector de cine Super 8, sonoro en perfecto estado, por un ordenador Spectrum o por algún periférico, impresora, etc. Mandar ofertas a Antonio Granados Martín. Larrinaga, 1, 1.°, Int. A. Bilbao (Vizcaya).

● VENDO ordenador Spectrum 48K, en perfecto estado, junto con toda una colección de revistas sobre el tema, manuales en castellano. Precio económico. Para más información llamar al Tel.: 6161609 de Madrid. Preguntar por Roberto.

● INTERCAMBIO todo tipo de información, trucos, ideas del ZX Spectrum. Contactar con Fco. Javier Hernández López. Las Postas, 36, 2.° Iz. Soria. Tel.: (975)211431.

● Si te gusta el mundo de los video-juegos, tengo una consola Atari CX2600, con transformador, 2 mandos de volante y un joystick casi nuevo, más 8 cartuchos. Para más información dirígirse al Tel.: (96)3855645. Preguntar por Laura o Carlos.

● SE busca gente que programe en Código Máquina para formar grupo de programación en Bilbao o alrededores. Llamar al Tel.: (94)4634791. Preguntar por Iñaki.

● VENDO ZX Spectrum 48K, con todo su contenido y en perfecto estado. Precio a convenir. Interesados llamar al Tel.: (93)3451305. Preguntar por Oscar.

● VENDO Spectrum 48K, con manual en castellano, juego de cables, transformador, cinta de Horizontes en castellano, interruptor de reset incorporado. Todo por 30 000 ptas. Llamar al Tel.: 2439472 de Madrid, preguntar por David Crespo.

● VENDO Spectrum 48K, completo (manuales, fuente de alimentación y cables) en perfecto estado y con garantía Investrónica, por sólo 29.000 ptas. Regalo un Joystick Kempston programable. Interesados llamar al Tel.: (983)255149. Preguntar por José.

● VENDO Spectrum 48K completo, con 4 meses de garantía. Precio: 26.700 ptas. Interesados escribir a Joaquín Alcázar. Laguna IV, 4, 2.° F. Tel.: (953)262691.

● VENDO Joystick y su interface por 5.000 ptas. Contactar con Isidro, llamanto al Tel.: 830124 de Guipúzcoa.

● VENDO Spectrum 16K, perfecto estado, con cables, fuente de alimentación, cinta Horizontes más ampliación a 48K, todo por 30.000 ptas. Llamar al Tel.: 244635 de Lérida. Preguntar por Miguel Angel. (A partir de las 4 tarde.)

● VENDO ZX Spectrum 48K, con cables,, cinta Horizontes, manual en castellano. Adjunto también interface programable y el correspondiente joystick. Todo ello en perfecto estado. Precio a convenir (aprox. 35.000 ptas.). Dirigirse a Tomás Saborido. Maudes, 16, 5.° Dr. 28003 Madrid. Tel.: 2335571.

● VENDO ZX Spectrum 48K, comprado en mar. 84. Incluye manual y todo lo necesario para su uso. Precio de 30.000 ptas. Interesados dirigirse a Jorge Ruiz Fernández. Goya, 2, 4.° G. Valladolid. O bien llamar al Tel.: (983) 239825.

● VENDO Consola de video-juegos con 8 compactos de juegos, marca TRQ-H21, nueva por 20.000 ptas. Interesados llamar al Tel.: (981)321062. Juan Carlos.

● VENDO ZX-81, con apliación de 16 K, (noviembre, 84), con manual de instrucciones inglés/castellano, cables, fuente de alimentación. Por sólo 15.000 ptas. (negociables). Interesados al Tlf. (93) 3338459. Preguntar por Madrid.

● QUISIERA contactar con usuarios del ZX Spectrum para intercambiar ideas, trucos, etc Escribir a Carles Jordi Fernández. Crta. a Bagá, 42, 2.°. Guardiola de Berguedá (BARCELONA).

● VENDO Commodore VIC-20. Datassette, 2 libros, 3 cintas, cables... por sólo 18.000 ptas. Escribir a José Manuel Barrena López. Canarias, 11, de Utrera (SEVILLA).

● QUISIERA comunicarme con usuarios del Spectrum 48 K, para formar un club de cambio de Software y revistas. También deseo ponerme en contacto con algún lector que pueda facilitarme las instrucciones de Fighter Pilot, Dictator y Paintbox. Pago fotocopias y gastos de envío. Escribir a Antonio Gallego Alberto. Avda. de Escalasitas, 56, ptal A, 6.° D. Palmas de Gran Canaria 38011.

"HOBBY SUERTE"

¡250.000 pts. en premios cada semana!

ESTE NUMERO PUEDE SIGNIFICAR UN FABULOSO REGALO PARA TI. **34379260**

¡Consulta a tu Spectrum!

Cada semana, Microhobby regala 70 premios entre sus lectores. La clave del premio es el número que figura en este cupón, en la esquina superior derecha.
Para saber si el número de tu ejemplar está premiado, debes introducirlo en tu Spectrum, utilizando para ello el programa «Hobby-Suerte». La cassette con este programa se ha entregado a los lectores junto con el número 15 de Microhobby Semanal. Si no posees esta cinta, puedes pedir una copia a un amigo o por carta a **Hobby Press, S.A., Apartado n.° 54062. Madrid,** incluyendo dentro del sobre 180 pts. en 3 sellos de Correos de 60 pts. cada uno. Este programa sirve para leer todos los números durante las **50 semanas** que dure este Concurso.

(Bases en el reverso)

Premios semanales

■ Primera Categoría

Un **Spectrum 48 k.** (o un Microdrive y un Interface 1, a elegir por el interesado).

■ Segunda Categoría

Una **Impresora GP 50 de Seikosha,** especialmente diseñada para Spectrum (2 premios).

■ Tercera Categoría

Un **Joystick con su interface** (3 premios).

■ Cuarta Categoría

Una **Suscripción a Microhobby Semanal** por un año (50 números. Si el lector premiado ya es suscriptor, podrá optar por prolongar su suscripción anual o un premio de Quinta Categoría) (14 premios).

■ Quinta Categoría

Una **Cinta de Programa,** a elegir entre un variado surtido de juegos, utilidades, etcétera (50 premios).

Hobby Press, S.A. garantiza que cada semana introduce al azar, entre todos los ejemplares que componen la edición, **setenta cupones** correspondientes a los premios aquí citados

"HOBBY SUERTE"

Instrucciones para concursar

Una vez introducido en memoria el programa **«Hobby-Suerte»,** aparecera en la panta lla la clásica máquina «tragaperras» de frutas.
A continuación, debes teclear el número que figura en esta tarjeta. Al pulsar «Enter», la «máquina» se pone en marcha y te hace saber si has sido agraciado con uno de los 70 premios semanales.

Muy importante: Puede ocurrir que, al introducir en el programa números al azar, alguno de ellos corresponda casualmente a un premio. También es posible, con los suficientes conocimientos de Basic, alterar el programa para que un número determinado aparezca como premiado.

Por todo ello, debemos aclarar los siguientes aspectos:

1. El único justificante para reclamar un premio determinado es la posesión del cupón con el número impreso en él.

2. Todos los números susceptibles de dar premio están registrados ante Notario.

3. Hobby Press, S.A. no se hace responsable de ningún otro cupón que no corresponda a los números previamente registrados. Tampoco se atenderán reclamaciones verbales que no vengan acompañadas por la posesión del cupón con el número premiado.

4. Cualquier lector puede solicitar de esta Editorial la comprobación de la entrega de los Premios semanales.

5. Hobby Press, S.A. se reserva el derecho a resolver según su criterio cualquier cuestión no prevista en las Bases de este Concurso

6. La reclamación de cualquier Premio de este Concurso caduca el día 30 de Junio de 1986

COMUNICACION DE PREMIO

(Enviar rellenado con letra clara y en sobre cerrado)

Nombre . . . . Edad . . . . . .
Apellidos . . . .
Domicilio . . . . Teléfono . . . .
Ciudad . . . . C P . . . . Provincia . . .
Categoría del Premio Obtenido . . . . Numero de Microhobby . . . . .

Si consideras que tu cupón tiene premio fotocopia o como medida de seguridad y envíalo por **Correo Certificado** a Hobby Press S A Apartado 54 062 de Madrid Por favor anticipanos todos estos datos por telefono, llamando a (91 654 32 11 En este mismo numero atenderemos cualquier consulta o duda sobre las Bases o a mecanica de «Hobby Suerte»

Envía este cupón por correo certificado a HOBBY PRESS. Apt. 54.062. Madrid

¡¡FABULOSO... HIPERSPORTS!!



☐ **SI**, deseo suscribirme a la Revista **Microhobby Semanal** durante un año (50 números), al precio actual de **4.250** pesetas, y recibir como regalo, sin gastos de envío por mi parte, una cinta original del programa «Hypersports», valorada en 2.100 pesetas.

(Debido a lo excepcional de esta oferta y hasta el 30 de septiembre, nos vemos obligados a suspender las modalidades de pago mediante reembolso o Tarjeta de Crédito. Por favor envíe un talón en un sobre cerrado, junto con este cupón o remita un giro postal. Gracias)



☐ **SI**, deseo recibir la(s) **Cinta(s) de Programas** que indico a continuación. Cada una de estas cintas lleva grabados los programas publicados por Microhobby Semanal durante cuatro numeros consecutivos (1 al 4, 5 al 8, 9 al 12, etcétera) y su precio es de 550 pesetas, más 75 ptas. por gastos de envio cada pedido (no por cada cinta).

Numeros del ___ al ___, (inclusive)	Números del ___ al ___, (inclusive)
Números del ___ al ___, (inclusive)	Números del ___ al ___, (inclusive)
Números del ___ al ___, (inclusive)	Numeros del ___ al ___, (inclusive)

(Para recibir numeros atrasados de Revistas, o estuches de encuadernación solicítelos mediante carta o tarjeta postal o por teléfono a los nums. (91) 733 50 12 y (91) 733 50 16. El precio de cada ejemplar es el mismo que figure en la portada y los estuches 495 ptas. cada uno. Los pedidos contra reembolso se incrementarán con 75 ptas. de gastos por cada envio)

OFERTAS VALIDAS SOLO PARA ESPAÑA

NOMBRE ________________ EDAD ________

APELLIDOS ________________

DOMICILIO ________________

CIUDAD ________________ PROVINCIA ________________

C. POSTAL ________ TELEFONO ________ PROFESION ________

Marco con una (X) en el casillero correspondiente la forma de pago que más me conviene

☐ Talón bancario adjunto a nombre de HOBBY PRESS S.A. ☐ Giro Postal N.° ___ ☐ Contra reembolso del primer N.°

TARJETA DE CREDITO ☐ VISA N.° ________ ☐ MASTER CHARGE N.° ________

Fecha de caducidad de la tarjeta ________ Firma

(cortar por la línea de trazos)

Franqueo
Postal

HOBBY PRESS, S.A.

Apartado de Correos

n.º **54.062** (Apartados Altos)

28080 MADRID



LD A,(BC)

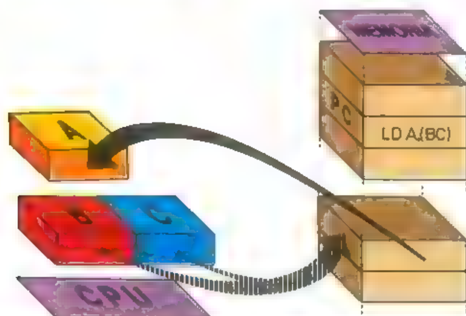
El contenido de 8 bits de la dirección de memoria especificada por el contenido del par BC es transferido al registro A.

Mnemónico: LD **Operandos:** A,(BC)

Formato binario:



Ciclos: 2
Estados: 7 (4,3)
Indicadores: ninguno



Instr.	Hex.	Dec.	Instr.	Hex.	Dec.
LD A,(BC)	0A	10	LD (BC),A	02	2
LD A, (DE)	1A	26	LD (DE),A	12	18

LD (BC),A

El contenido del registro A es transferido a la dirección de memoria especificada por el contenido del par BC.

Mnemónico: LD **Operandos:** (BC),A

Formato binario:



Ciclos: 2
Estados: 7 (4,3)

Indicadores: ninguno

Ejemplo:

Si el contenido del par BC es 3000H, y el contenido del registro A es 7FH, después de ejecutar la instrucción: LD (BC),A
resultará que la dirección de memoria 3000H contiene 7FH.

Menaje de pantalla

La forma de llamada es RANDOMIZE USR n+d, siendo n la dirección donde se ubicará la rutina y d el desplazamiento de la subrutina que queremos utilizar para operar con los ficheros de imagen de 2 pantallas, la del sistema y la de trabajo, situada a partir de la dirección 32000.

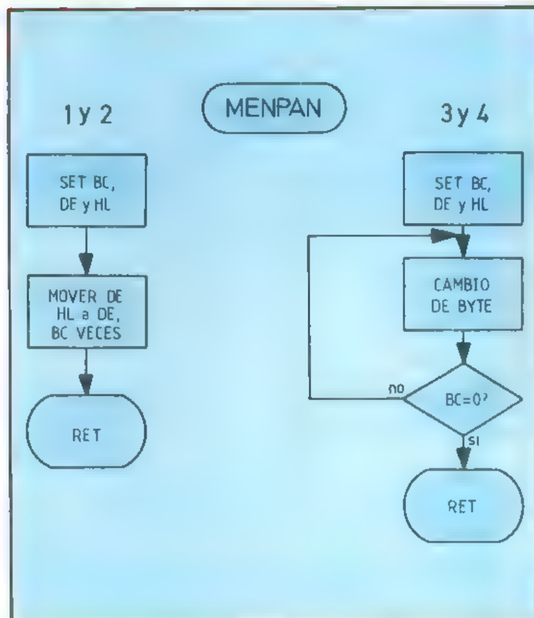
El valor d puede ser 0 (almacenamiento en la pantalla de trabajo), 12 (recuperación de la pantalla de trabajo), 24 (intercambio de ambas pantallas), 47 (mezcla de ambas pantallas).

Para d=47 se puede fijar el modo de mezclado usando la instrucción POKE n+57, códigos 174 (OVER 1 "XOR (HL)"), 182 (OVER 0 "OR (HL)"), 166 (intersección "AND (HL)"), 126 (intercambia el archivo de imagen "LD A,(HL)"), o 47 (INVERSE 1).

Funcionamiento:

Para d=0 y d=12 se carga la dirección inicial de una pantalla en el par HL y la longitud en el par BC, y se transfiere a una zona de memoria cuyo comienzo está especificado por el par DE.

Para d=24 y d=47 se repite un bucle que barre los ficheros de imagen de ambas pantallas, intercambiándolos o mezclándolos.



LD A,(DE)

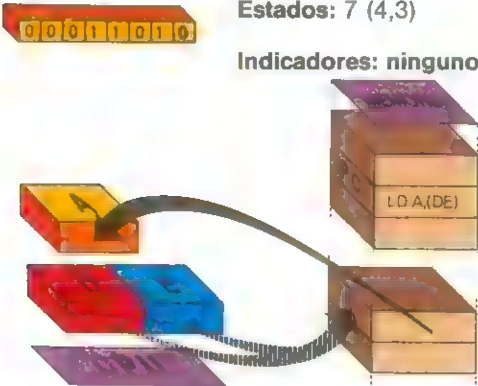
El contenido de 8 bits de la dirección de memoria especificada por el contenido del para DE es transferido al registro A.

Mnemónico: LD **Operandos:** A,(DE)

Formato binario:

Ciclos: 2
Estados: 7 (4,3)

Indicadores: ninguno



LD (DE),A

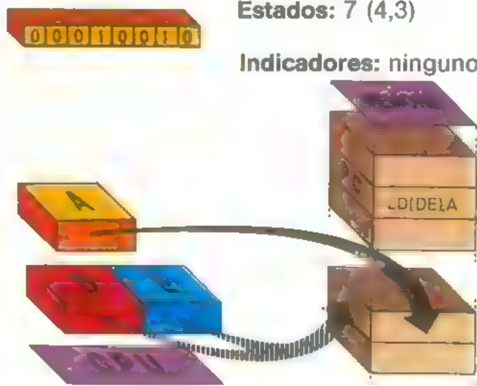
El contenido del registro A es transferido a la dirección de memoria especificada por el contenido del par DE.

Mnemónico: LD **Operandos:** (DE),A

Formato binario:

Ciclos: 2
Estados: 7 (4,3)

Indicadores: ninguno



El microprocesador Z80 A tiene registros cuya característica es la de acceder a ellos para almacenamiento de **datos temporales** para poder realizar operaciones con ellos sin necesidad de utilizar memoria RAM externa. Existen dos juegos de registros de propósito general pudiéndose reservar un juego de ellos además del AF para el manejo de una rutina de acción inmediata.

1. El Acumulador
2. El par de registros HL
3. Los pares de registros BC y DE
4. Los registros indexados IX e IY
5. El puntero de pila o SP
6. Los registros especiales:
   - Registro de banderas o Flags
   - Registro de interrupciones
   - Registro de refresco de memoria

1. El Acumulador:

Es el registro más utilizado ya que realiza y contiene el resultado de las operaciones lógicas y aritméticas con 8 bits. Las operaciones que pueden realizarse con el acumulador son: transferencias, suma, resta, AND y OR lógicos, XOR (or exclusivo lógico), comparaciones y complementación a 1 y a 2.

2. El par HL:

Es el par de registros más versátil de todos los que contiene el Z80 A, utilizado normalmente para contener las direcciones de memoria que se van utilizando durante el transcurso de una rutina, ya que algunas operaciones con los otros pares (BC y DE) no son ejecutables.

3. Los pares BC y DE:

Suelen utilizarse como pares auxiliares del HL en instrucciones que manipulan bloques tales como LDI, LDIR, etc.

4. Registros indexados IX e IY:

Los registros índice se utilizan como base para apuntar a una región de memoria de donde se va


```
 10 ;*  MENAJE DE PANTALLAS *
 20 ;
 30 ;
 40        ORG    60000    ;RUTINA REUBICABLE
 50 ;
 60 ; ALMACENAMIENTO DE PANTALLA
 70 .
 80 START1 LD     HL,16384 ;Com. de la pantalla
 90        LD     DE,32000 ;Dir. de la pant. 2
100        LD     BC,6912  ;Longit. de la pant.
110        LDIR   ;        ;Almacena la pantalla
120        RET
130 ,
140 ; RECUPERACION DE PANTALLA
150 START2 LD     HL,32000 ;Dir. de la pant. 2
160        LD     DE,16384 ;Comienzo de la pant.
170        LD     BC,6912  ;Longit. de la pant.
180        LDIR   ;        ;Recupera la pantalla
190        RET
200 ;
210 ; INTERCAMBIO DE PANTALLAS
220 START3 LD     HL,32000 ;Dir. de la pant. 2
230        LD     DE,16384 ;Comienzo de la pant.
240        LD     BC,6912  ;Long. de la pantalla
250 BUCLE1 LD     A,(DE)   ;Intercambia el
260        EX     AF,AF'   ;contenido de
270        LD     A,(HL)   ;la pantalla con la
280        LD     (DE),A   ; pantalla almacenada
290        EX     AF,AF'
300        LD     (HL),A
310        INC    DE       ;Pantalla 1
320        INC    HL       ;Pantalla 2
330        DEC    BC       ;Longitud de pantalla
340        LD     A,B      ;
350        OR     C        ;Comprueba si BC=0
360        JR     NZ,BUCLE1;si no, repite BUCLE1
370        RET
380 ;
390 ;  MEZCLA DE PANTALLAS
400 START4 LD     HL,32000 ;Dir. de la pant. 2
410        LD     DE,16384 ;Com. del DISP.FILE
420        LD     BC,6144  ;Long. DISP.FILE
430 BUCLE2 LD     A,(DE)   ;Cont. del DISP FILE
440 MODO   XOR    (HL)     ;XOR con la pant. 2
450        LD     (DE),A   ;Result. al DISP.FILE
460        INC    DE       ;DISPLAY FILE
470        INC    HL       ;Segunda pantalla
480        DEC    BC       ;Long. del DISP.FILE
490        LD     A,B      ;
500        OR     C        ;Comprueba si BC=0
510        JR     NZ,BUCLE2;si no, repite BUCLE2
520        RET
```

```
10 DATA "21 00 40 11 00 7D 01 00",240
20 DATA "1B ED B0 C9 21 00 7D 11",816
30 DATA "00 40 01 00 1B ED B0 C9",706
40 DATA "21 00 7D 11 00 40 01 00",240
50 DATA "1B 1A 08 7E 12 08 77 13",351
60 DATA "23 0B 78 B1 20 F3 C9 21",852
70 DATA "00 7D 11 00 40 01 00 18",231
80 DATA "1A AE 12 13 23 0B 78 B1",580
90 DATA "20 F6 C9                ",479
```

La función **USR** del Basic del ZX Spectrum es como el cordón umbilical que une el Basic en sí, con los programas escritos en código máquina.

Realiza además otra función, cuando el argumento es de tipo cadena, que nos da la dirección de comienzo de los caracteres **UDG** (Gráficos definibles por el usuario).

Con una expresión numérica, el BASIC hará una llamada a una subrutina en código máquina que comience en la dirección indicada por el valor de la expresión.

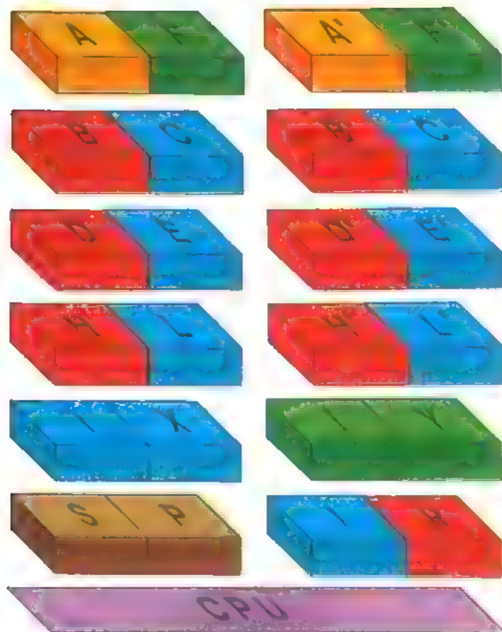
En la subrutina debemos preservar el par de registros IY, que es el puntero para las variables del sistema, y debe apuntar siempre a la variable **ERR-NR**, dirección 23610 (5C3AH).

Debemos también preservar el par de registros **HL'**, que contiene información necesaria para el calculador del BASIC.

Podemos, además, conocer la dirección de comienzo de la subrutina, que está en el par de registros BC, dato necesario para reubicación y manejo de memoria.

Llamada a una subrutina en código máquina
Dirección de llamada
Parámetros numéricos con POKEs
Parámetros numéricos con REM
Parámetros numéricos en expresión
Valor de retorno

Por otra parte, la función USR devuelve el valor en decimal del par de registros **BC.**, muy útil para usar con variables numéricas, por ejemplo, **LET num=USR nn,** donde se llama a una subrutina que comienza en la dirección nn, y al volver, la variable numérica «num» tiene el valor decimal del par **BC.**



a tomar o almacenar un dato. Se incluye un byte adicional para especificar un desplazamiento de esta base.

5. El puntero de pila SP:

La pila o stack está organizada de forma que el último dato que entra en la misma es el primero que sale. Esta organización permite el anidamiento ilimitado de rutinas.

6. Registros especiales:

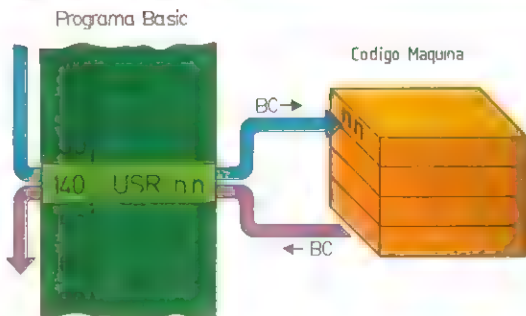
— Registro de indicadores o Flags (F): indica las condiciones que se producen al realizar las operaciones en 8 y 16 bits.

— Registro de interrupciones I: Se utiliza para ejecutar cualquier subrutina como respuesta a una interrupción en modo IM2.

— Registro de refresco de memoria R: el dato del contador de refresco se coloca en la parte baja del bus de direcciones junto con una señal de control de refresco proporcionada por la CPU, mientras ésta busca y decodifica la instrucción.

Para pasar a su vez parámetros a la subrutina, podemos utilizar 3 sistemas:

- **POKE**ando los valores numéricos en las direcciones determinadas.
- Colocándolos en una instrucción **REM,** en la siguiente línea después de la función, que no provoca errores de sintaxis, cuya dirección de comienzo está en la variable del sistema **NEXTLIN,** dirección 23637 (5C55H).
- Usando USR en una expresión que conlleve el almacenamiento de los parámetros en el **Stack** del calculador BASIC, teniendo en cuenta la jerarquía de la expresión.
  Ej.: RANDOMIZE 1 + a * USR nn
  En este caso, el Basic chequea la expresión, y carga en el Stack los valores 1, y el de la variable «a», y antes de realizar las operaciones ejecuta la llamada al código máquina, puesto que por tener mayor prioridad, ha de realizar primero la multiplicación, en la que USR nn es el multiplicador, y el resultado final de la expresión se usará para el RANDOMIZE en sí, almacenándolo en la variable de sistema **SEED,** dirección 23670 (5C76H).



Cuando se trabaja con el **Interface 1,** sólo se puede utilizar con las instrucciones RANDOMIZE y LET, puesto que garantiza la correcta paginación de la ROM principal, contra otras instrucciones, especialmente IF USR nn, que pueden dejar el sistema completamente «colgado».



El microprocesador Z80 A tiene registros cuya característica es la de acceder a ellos para almacenamiento de **datos temporales** para poder realizar operaciones con ellos sin necesidad de utilizar memoria RAM externa. Existen dos juegos de registros de propósito general pudiéndose reservar un juego de ellos además del AF para el manejo de una rutina de acción inmediata.

1. **El Acumulador**
2. **El par de registros HL**
3. Los pares de registros BC y DE
4. Los registros indexados IX e IY
5. **El puntero de pila o SP**
6. **Los registros especiales:**
   - Registro de banderas o Flags
   - Registro de interrupciones
   - Registro de refresco de memoria

1. El Acumulador:

Es el registro más utilizado ya que realiza y contiene el resultado de las operaciones lógicas y aritméticas con 8 bits. Las operaciones que pueden realizarse con el acumulador son: transferencias, suma, resta, AND y OR lógicos, XOR (or exclusivo lógico), comparaciones y complementación a 1 y a 2.

2. El par HL:

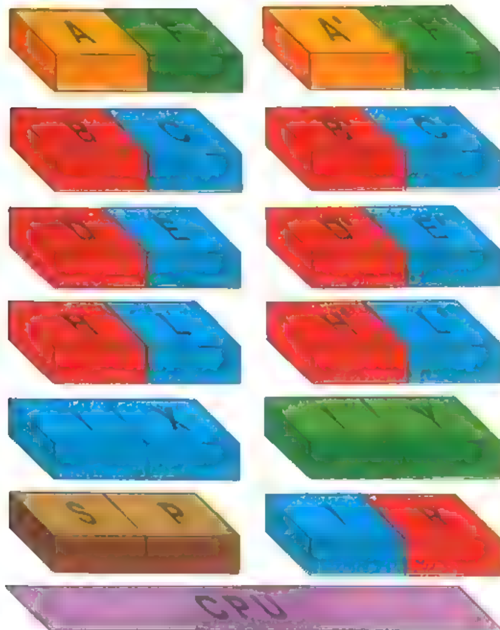
Es el par de registros más versátil de todos los que contiene el Z80 A, utilizado normalmente para contener las direcciones de memoria que se van utilizando durante el transcurso de una rutina, ya que algunas operaciones con los otros pares (BC y DE) no son ejecutables.

3. Los pares BC y DE:

Suelen utilizarse como pares auxiliares del HL en instrucciones que manipulan bloques tales como LDI, LDIR, etc.

4. Registros indexados IX e IY:

Los registros índice se utilizan como base para apuntar a una región de memoria de donde se va



a tomar o almacenar un dato. Se incluye un byte adicional para especificar un desplazamiento de esta base.

5. El puntero de pila SP:

La pila o stack está organizada de forma que el último dato que entra en la misma es el primero que sale. Esta organización permite el anidamiento ilimitado de rutinas.

6. Registros especiales:

— Registro de indicadores o Flags (F): indica las condiciones que se producen al realizar las operaciones en 8 y 16 bits.

— Registro de interrupciones I: Se utiliza para ejecutar cualquier subrutina como respuesta a una interrupción en modo IM2.

— Registro de refresco de memoria R: el dato del contador de refresco se coloca en la parte baja del bus de direcciones junto con una señal de control de refresco proporcionada por la CPU, mientras ésta busca y decodifica la instrucción.

La función USR



La función **USR** del Basic del ZX Spectrum es como el cordón umbilical que une el Basic en sí, con los programas escritos en código máquina.

Realiza además otra función, cuando el argumento es de tipo cadena, que nos da la dirección de comienzo de los caracteres **UDG** (Gráficos definibles por el usuario).

Con una expresión numérica, el BASIC hará una llamada a una subrutina en código máquina que comience en la dirección indicada por el valor de la expresión.

En la subrutina debemos preservar el par de registros IY, que es el puntero para las variables del sistema, y debe apuntar siempre a la variable **ERR-NR,** dirección 23610 (5C3AH).

Debemos también preservar el par de registros **HL'**, que contiene información necesaria para el calculador del BASIC.

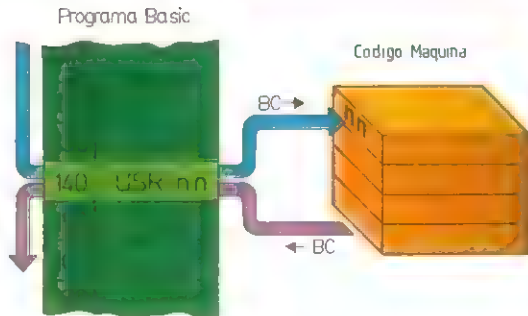
Podemos, además, conocer la dirección de comienzo de la subrutina, que está en el par de registros BC, dato necesario para reubicación y manejo de memoria.

Llamada a una subrutina en código máquina
Dirección de llamada
Parámetros numéricos con POKEs
Parámetros numéricos con REM
Parámetros numéricos en expresión
Valor de retorno

Por otra parte, la función USR devuelve el valor en decimal del par de registros **BC.,** muy útil para usar con variables númericas, por ejemplo, **LET num=USR nn,** donde se llama a una subrutina que comienza en la dirección nn, y al volver, la variable numérica «num» tiene el valor decimal del par **BC.**

Para pasar a su vez parámetros a la subrutina, podemos utilizar 3 sistemas:

- **POKE**ando los valores numéricos en las direcciones determinadas.
- Colocándolos en una instrucción **REM**, en la siguiente línea después de la función, que no provoca errores de sintaxis, cuya dirección de comienzo está en la variable del sistema **NEXTLIN**, dirección 23637 (5C55H).
- Usando USR en una expresión que conlleve el almacenamiento de los parámetros en el **Stack** del calculador BASIC, teniendo en cuenta la jerarquía de la expresión.
  Ej.: RANDOMIZE 1 + a * USR nn
  En este caso, el Basic chequea la expresión, y carga en el Stack los valores 1, y el de la variable «a», y antes de realizar las operaciones ejecuta la llamada al código máquina, puesto que por tener mayor prioridad, ha de realizar primero la multiplicación, en la que USR nn es el multiplicador, y el resultado final de la expresión se usará para el RANDOMIZE en sí, almacenándolo en la variable de sistema **SEED**, dirección 23670 (5C76H).



Cuando se trabaja con el **Interface 1**, sólo se puede utilizar con las instrucciones RANDOMIZE y LET, puesto que garantiza la correcta paginación de la ROM principal, contra otras instrucciones, especialmente IF USR nn, que pueden dejar el sistema completamente «colgado».

LD A,(BC)

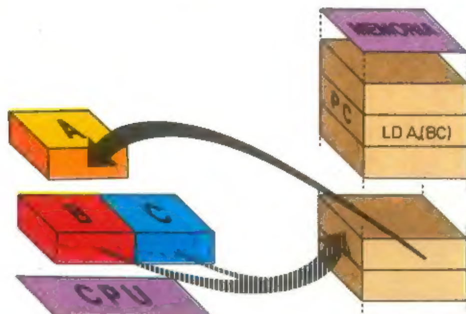
El contenido de 8 bits de la dirección de memoria especificada por el contenido del par BC es transferido al registro A.

Mnemónico: LD   **Operandos:** A,(BC)

Formato binario:



Ciclos: 2
Estados: 7 (4,3)
Indicadores: ninguno



Instr.	Hex.	Dec.	Instr.	Hex.	Dec.
LD A,(BC)	0A	10	LD (BC),A	02	2
LD A, (DE)	1A	26	LD (DE),A	12	18

LD (BC),A

El contenido del registro A es transferido a la dirección de memoria especificada por el contenido del par BC.

Mnemónico: LD   **Operandos:** (BC),A

Formato binario:



Ciclos: 2
Estados: 7 (4,3)

Indicadores: ninguno

Ejemplo:

Si el contenido del par BC es 3000H, y el contenido del registro A es 7FH, después de ejecutar la instrucción: LD (BC),A
resultará que la dirección de memoria 3000H contiene 7FH.

LD A,(DE)

El contenido de 8 bits de la dirección de memoria especificada por el contenido del para DE es transferido al registro A.

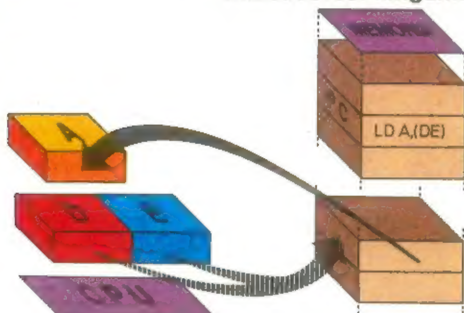
Mnemónico: LD **Operandos:** A,(DE)

Formato binario:



Ciclos: 2
Estados: 7 (4,3)

Indicadores: ninguno



LD (DE),A

El contenido del registro A es transferido a la dirección de memoria especificada por el contenido del par DE.

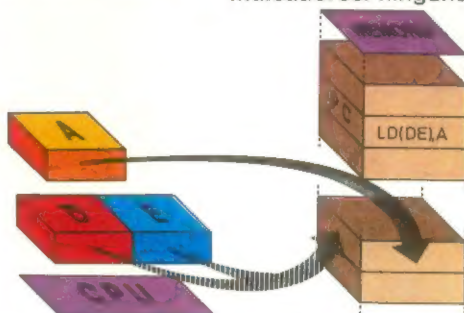
Mnemónico: LD **Operandos:** (DE),A

Formato binario:



Ciclos: 2
Estados: 7 (4,3)

Indicadores: ninguno



Menaje de pantalla

La forma de llamada es RANDOMIZE USR n+d, siendo n la dirección donde se ubicará la rutina y d el desplazamiento de la subrutina que queremos utilizar para operar con los ficheros de imagen de 2 pantallas, la del sistema y la de trabajo, situada a partir de la dirección 32000.

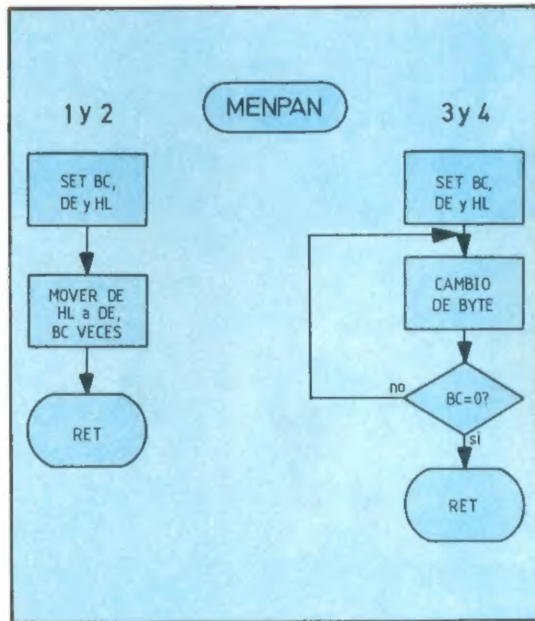
El valor d puede ser 0 (almacenamiento en la pantalla de trabajo), 12 (recuperación de la pantalla de trabajo), 24 (intercambio de ambas pantallas), 47 (mezcla de ambas pantallas).

Para d=47 se puede fijar el modo de mezclado usando la instrucción POKE n+57, códigos 174 (OVER 1 "XOR (HL)"), 182 (OVER 0 "OR (HL)"), 166 (intersección "AND (HL)"), 126 (intercambia el archivo de imagen "LD A,(HL)"), o 47 (INVERSE 1).

Funcionamiento:

Para d=0 y d=12 se carga la dirección inicial de una pantalla en el par HL y la longitud en el par BC, y se transfiere a una zona de memoria cuyo comienzo está especificado por el par DE.

Para d=24 y d=47 se repite un bucle que barre los ficheros de imagen de ambas pantallas, intercambiándolos o mezclándolos.



```
10 ;*  MENAJE DE PANTALLAS *
20 ;
30 ;
40         ORG    60000     ;RUTINA REUBICABLE
50 ;
60 ;  ALMACENAMIENTO DE PANTALLA
70 ;
80 START1 LD      HL,16384 ;Com. de la pantalla
90        LD      DE,32000 ;Dir. de la pant. 2
100       LD      BC,6912  ;Longit. de la pant.
110       LDIR    ;        ;Almacena la pantalla
120       RET
130 ;
140 ; RECUPERACION DE PANTALLA
150 START2 LD     HL,32000 ;Dir. de la pant. 2
160       LD      DE,16384 ;Comienzo de la pant.
170       LD      BC,6912  ;Longit. de la pant.
180       LDIR    ;        ;Recupera la pantalla
190       RET
200 ;
210 ; INTERCAMBIO DE PANTALLAS
220 START3 LD     HL,32000 ;Dir. de la pant. 2
230       LD      DE,16384 ;Comienzo de la pant.
240       LD      BC,6912  ;Long. de la pantalla
250 BUCLE1 LD     A,(DE)   ;Intercambia el
260       EX      AF,AF'   ;contenido de
270       LD      A,(HL)   ;la pantalla con la
280       LD      (DE),A   ; pantalla almacenada
290       EX      AF,AF'
300       LD      (HL),A
310       INC     DE       ;Pantalla 1
320       INC     HL       ;Pantalla 2
330       DEC     BC       ;Longitud de pantalla
340       LD      A,B      ;
350       OR      C        ;Comprueba si BC=0
360       JR      NZ,BUCLE1;si no, repite BUCLE1
370       RET
380 ;
390 ;  MEZCLA DE PANTALLAS
400 START4 LD     HL,32000 ;Dir. de la pant. 2
410       LD      DE,16384 ;Com. del DISP.FILE
420       LD      BC,6144  ;Long. DISP.FILE
430 BUCLE2 LD     A,(DE)   ;Cont. del DISP.FILE
440 MODO  XOR     (HL)     ;XOR con la pant. 2
450       LD      (DE),A   ;Result. al DISP.FILE
460       INC     DE       ;DISPLAY FILE
470       INC     HL       ;Segunda pantalla
480       DEC     BC       ;Long. del DISP.FILE
490       LD      A,B      ;
500       OR      C        ;Comprueba si BC=0
510       JR      NZ,BUCLE2;si no, repite BUCLE2
520       RET
```

```
10 DATA "21 00 40 11 00 7D 01 00",240
20 DATA "1B ED B0 C9 21 00 7D 11",816
30 DATA "00 40 01 00 1B ED B0 C9",706
40 DATA "21 00 7D 11 00 40 01 00",240
50 DATA "1B 1A 08 7E 12 08 77 13",351
60 DATA "23 0B 78 B1 20 F3 C9 21",852
70 DATA "00 7D 11 00 40 01 00 18",231
80 DATA "1A AE 12 13 23 0B 78 B1",580
90 DATA "20 F6 C9                ",479
```